

Font Management in macOS, by Kurt Lang

Last updated January 24, 2019

This article deals with font usage in El Capitan 10.11.x through Mojave 10.14.x. Its main purpose is to show you where fonts are located on your system and which can be safely removed. The idea is to keep your font list as small as possible to avoid font conflicts (font conflicts are explained in Section 13). This article will benefit prepress operators and graphic designers the most, but can clear up font issues for most general users as well.

It should be noted that this article is written around the assumption that you are using English as your primary language. The minimum required fonts will be very different for other languages.

To view the older version of this article and access information back to OS X, 10.3, click [here](#).

If you find this article useful, please consider making a donation via [PayPal](#). Contributions to user account retoucher@jklstudios.com

Thank you.

What changed in this article update?

- A new note regarding Times being added to Excel.
- Fixed the very outdated list of the last section on reinstalling fonts.
- Minor changes on the removal of system fonts. Mainly regarding El Capitan vs. later OS releases.
- Office 365 font removal instructions. **Cloud font listing issue fixed!** This also works for Office 2019.
- Font review updates.

Table of contents

Preface: Understanding terminology used in this article.

Sections:

1. Required fonts.
2. Enabling and disabling System Integrity Protection
3. Known issues with disabled fonts.
4. Fonts installed by Microsoft Office.
5. What to do with Suitcase.
6. Controlling Helvetica fonts for prepress.
7. Font folder locations. Using Font Book.
8. Various other fonts you can move or remove.
9. Do you need a font manager?
10. What is a font suitcase and why do we have them?
11. FFIL and LWFN. What are they and which is which?
12. External vs. internal font names.
13. What is a font conflict?
14. Are you a good font, or a bad font?
15. Fonts you can and cannot use in macOS.
16. Font manager reviews.
17. How to handle font caches.
18. Font 911. What to do in font emergencies.
19. Determining if fonts are causing problems with applications.
20. Reinstalling your macOS supplied fonts.

Preface

I first want to mention the notation of file locations. By 'notation' I am referring to the path name. This should help novice computer users and those unfamiliar with standard notation to learn how to navigate to the folders mentioned throughout this article.

I can't tell you exactly what the path to your home account looks like (since I don't know your short user name), so here are some handy notes of reference.

A *file specification* is the entire path from the root of the volume it resides on to the end of the file name. For example, here is the file specification for the Terminal application:

`/Applications/Utilities/Terminal`

This is known as a *hierarchical file specification* in geek terminology, but it's called a canonical filename for short.

`/` The beginning forward slash (as in the example to the Terminal application) of a file specification is always the root level of your boot volume.

`~/` The tilde-forward slash pair is always your home directory (folder), i.e., the home folder of the current user login session.

So in most cases, the path to the Fonts folder in your home user account would be `~/Library/Fonts/`. Which, if you start by double clicking the icon of the boot drive on the desktop, the path can also be presented as `/Users/your_user_account/Library/Fonts/`.

The following words: program, application or app all have the same meaning. I use them interchangeably throughout this article.

Required fonts

This section examines each of the various macOS releases (El Capitan 10.11.x through Mojave 10.14) and provides the recommended minimum list of the fonts to be stored in the System folder for that particular release of the operating system in order for it and most third party applications to run properly. These lists also include the fonts most needed for the web, iLife and iWork. The fonts listed should always be active on your Macintosh for macOS and should not be removed.

Note that this first part of Section 1 covers only fonts required in the `/System/Library/Fonts/` folder. There is also a root `/Library/Fonts/` folder with its own set of required fonts, which will be addressed in the second half of Section 1.

From the font lists below, the San Francisco font set is used throughout the system font display purposes. This set, and the fonts HelveticaNeueDeskInterface.ttc and LucidaGrande.ttc must be present for the Finder and OS installed application menus to work.

All other fonts in the `/System/Library/Fonts/` folder (that are not included in the lists below by release level) can be removed. You will need Administrative access to delete fonts from this folder. It is advised to save them for future use. Create a new folder on your hard drive and copy them there first. If there are any removed fonts you want to use for a project at a later date, they can always be activated with [Font Book](#), [Suitcase Fusion](#), [FontAgent](#), [FontExplorer X Pro](#), [TypeDNA](#), or other font manager.

There are many fonts you can still remove from the `/System/Library/Fonts/` folder. See section 6 on how to permanently remove Apple's supplied versions of Helvetica and Helvetica Neue in El Capitan, 10.11 if this is important for you. El Capitan 10.11 through Mojave 10.14 added System Integrity Protection to the OS, making it yet more difficult to remove unneeded fonts, but it can still be done.

Other than those fonts the OS absolutely requires to function, when it comes to the fonts you prefer to have on your system there is no right or wrong list. My idea of required fonts is based on years in prepress. So like most shops, the number of fonts throughout the system is kept to a minimum in the effort to make sure you will never have a conflict with the fonts a client sends with their project. Actually, the prepress and full service printers I've worked and freelanced for usually have a much shorter list than even those presented here. Sometimes the barest minimum of fonts they can get away with and still have the OS function. Such shops normally have no unnecessary software installed on their work stations; just what's needed to get production work done.

This bare minimum setup has some advantage, but you will then be missing many fonts commonly used on the web. What then happens is that your browser ends up substituting the missing fonts with whatever font is available. The result is that web pages will display so badly at times that it can be difficult (or even impossible) to navigate them. The bare minimum setup also lacks many fonts that Apple supplied applications require to operate. So for most users having only the bare minimum fonts on your system is not recommended.

You can find many different web sites telling you what the minimum font installation for each macOS release should be. Each site has its own reasons for including some fonts that I do not, and others don't include fonts I think should be active. My main decision making was to run every application the OS ships with and many major third party applications, seeing what wouldn't work if a particular font were missing. The end result is the list of fonts you find here. It's a compromise between the Spartan set most prepress shops use, and what a more fully functional OS needs along with proper display of web pages.

Special Notes About Section 1 - Presented in no particular order as each OS release changes the rules a bit. Hopefully each is organized into its own paragraph, but no promises.

Readers who have followed this article for some time will note that Times and Symbol have been added to the required font lists. They were excluded before since this article was originally intended as a guide for prepress, when the article was also much shorter in length. So the lists have been modified to represent what the majority of macOS users should have in their `/System/Library/Fonts/` folder, rather than the leaning towards the needs of prepress. For that reason, Courier has been added back into the minimum font lists for the System folder. As with Times and Symbol, remove Courier if it interferes with your need to use a PostScript version.

Users should be aware that not all font managers, and possibly other utilities, will list font names exactly as you see them here. For example, Suitcase Fusion's interface lists Keyboard and Helvetica Neue Desk UI as having a period preceding their names (those come from the font's internal names). Font Book also hides some fonts in its listings from the user in Snow Leopard and later, such as LastResort and Keyboard. But you shouldn't be removing those fonts anyway.

If you haven't already, purchase Adobe's or Linotype's new OpenType PostScript Helvetica fonts if you prefer, or require PostScript fonts for your output. They do not conflict with Apple's Helvetica fonts, so you don't have to fight with the OS supplied fonts as to which ones are active. Use Type 1 PostScript when you have to accurately reproduce a standing older project (see section 6 if this applies to you).

One thing to be aware of when you disable Apple's Helvetica.ttc and HelveticaNeue.ttc, is that you are disabling quite a few fonts. This is because a .ttc is a suitcase which can contain any number of individual fonts. The following list is based on High Sierra.

Helvetica: Regular, Bold, Bold Oblique, Light, Light Oblique, Oblique

Helvetica Neue: Regular, Bold, Bold Italic, Italic, Light, Light Italic, Medium, Medium Italic, Thin, Thin Italic, UltraLight, UltraLight Italic, Condensed Black, Condensed Bold

Beginning with El Capitan, Apple had almost released Helvetica and Helvetica Neue back to the user. Adobe, Microsoft and possibly other third party vendors have not. See the minimum font list for El Capitan for more information. In High Sierra, it appears Apple has stopped using Helvetica and Helvetica Neue for the OS entirely.

Apple's Grapher program is not something normally used in prepress, which relies on the fonts Times and Symbol. As clients frequently use other versions of Times and Symbol, the Apple supplied versions can be excluded from the lists below if you need them out of the way. See section 3 for more on Grapher.

A note on the MM fonts in the `/System/Library/Fonts/` folder. Since Lion, the Mac OS has continued to install these Multiple Master fonts. Also since Lion, a Terminal command named `fontrestore` has existed, which attempts move all third party fonts out of the System, main Library, and the active user account Fonts folders. Run with the `-n` option, it pretends to go through the steps without actually doing anything; though it still does create the folder `Fonts (Removed)` in the main `/Library/`, the `/System/Library/` and the active user account folders. When the Terminal command is run, it produces this "error" message:

These fonts are not part of the default system install. They would have been removed to 'Fonts (Removed)':

```
/System/Library/Fonts/HelvetTMM  
/System/Library/Fonts/TimesTMM
```

The message is wrong since a default install of El Capitan through Mojave *will* install these files. Font Book runs the Unix command `fontrestore` under the option `Restore Standard Fonts`. When run, it does indeed remove the MM fonts. Proof enough for me they're dead. For this reason, they are no longer included in the list of required fonts in El Capitan or later. The initial purpose for these fonts was to duplicate the Adobe Reader's built in MM fonts for use in Preview. These MM fonts no longer exist in the Adobe Reader, and it appears Apple has followed suit, but hasn't cleaned up the OS installers.

It should also be noted that this command *does not* restore all fonts installed by macOS you may have removed from the System or root Library folders. What *does* restore are System and root Library fonts you may have removed that also exist in the hidden Recovery partition. This is a very incomplete set. Some will come back, but most won't. The command also removes fonts which are not part of the macOS original installation. As such, it "restores" the `/System/Library/Fonts/` folder, the `/Library/Fonts/` folder to a state which only includes the fonts provided with macOS. The active user account Fonts folder gets emptied out.

To make your user account Library folder permanently visible, open your user account by double clicking the icon of the house within the Users folder. It must be the active folder in the Finder in order for this to work. The fastest way to get to this folder is to be on the desktop (so **Finder** is shown as the active app next to the Apple logo at the upper left) and press `Command+Shift+H`. Then choose **View > Show View Options**, or press `Command+J`. There will be a check at the bottom labeled `Show Library Folder`. If the correct user account folder is not open and selected, you will not see this check box.

Required fonts in the `/System/Library/Fonts/` folder.

The following lists, arranged by the release level of Apple's desktop OS, are the minimum recommended fonts. They represent the minimum number of fonts that allow all macOS supplied apps, and most third party apps to work. The latter being limited to what I can test. *Always* save copies of *all* installed macOS fonts before proceeding.

If there are apps you use that will not launch after reducing your fonts to these lists, enable the copied fonts one at a time with your font manager (or just temporarily move or copy them into the Fonts folder of your user account) and test the app again. Keep adding until the app launches successfully. Permanently add that font back to the system. As an example, some of the Adobe CS6 and CC 2018 apps will not launch if Helvetica is missing. Or, parts of them will not display properly. Such testing is sometimes more involved than that. For instance, the early release of Microsoft Office 2016 would not reliably launch unless HelveticaNeue.dfont was specifically in the `/System/Library/Fonts/` folder.

Any time you manually remove fonts, you should clear the font cache files from the system. Remove all fonts first, then see section 17 for instructions. The method using Terminal at the bottom of that section is the easiest. If you use Font Book, you should reset its database (section 7)

In the attempt of being verbose, the System font lists were getting rather lengthy. To greatly shorten them, I've condensed the San Francisco fonts to one line. All fonts that start with SF in the `/System/Library/Fonts/` folder stay.



macOS, 10.14 Mojave

Apple has again expanded the number of San Francisco fonts in macOS. There are now 79 in Mojave. The required fonts for Mojave are the same as High Sierra, plus the new, additional SF fonts.

Apple continues the orphanage of a handful of its own supplied fonts. Those currently being Athelas.ttc, Iowan Old Style.ttc, Marion.ttc, Seravek.ttc and SuperClarendon.ttc. As before, you can use these fonts in any third party app, but they will not appear in any app written and supplied by Apple. The fix is the same as in High Sierra. You must retrieve the same named fonts from Yosemite and replace those installed by Mojave.

To remove fonts from the `/System/Library/Fonts/` folder in Mojave, [disable System Integrity Protection](#). With SIP disabled, you can put the system fonts you don't want in the trash and empty it. No need to first restart the Mac before the OS will let you do that. Re-enable SIP when you've finished removing the fonts.

Or, you can startup to another bootable drive and remove system fonts from the non-startup drive without disabling SIP at all. Only your admin credentials are required.

There's an issue now with most font managers, including Font Book. If you try to reactivate copies of any removed fonts from the `/System/Library/Fonts/` folder with your font manager, most will not appear in its list of imported fonts. You may also get a message about not having permission to read the fonts.

This only happens with fonts you've removed from the System folder, and only with some, but not all font managers. Font Book, Suitcase Fusion and FontExplorerX Pro all have problems with these fonts. FontAgent is unaffected.

Fortunately, there's a simple solution. If you do need any fonts previously in the System folder activated, you can manually place them in the Fonts folder of your user account.

The minimum fonts recommended for Mojave in the `/System/Library/Fonts/` folder are:

- Apple Color Emoji.ttc
- AppleSDGothicNeo.ttc
- Avenir.ttc
- Courier.dfont
- Geneva.dfont
- Helvetica.ttc
- HelveticaNeue.ttc
- HelveticaNeueDeskInterface.ttc
- Keyboard.ttf
- LastResort.otf
- LucidaGrande.ttc
- Menlo.ttc
- Monaco.dfont
- PingFang.ttc

SF... (all San Francisco fonts)
Symbol.ttf
Times.ttc
ZapfDingbats.ttf
ヒラギノ角ゴシック W3.ttc

Fonts updated in 10.14.2

Apple Color Emoji.ttc
Helvetica.ttc
HelveticaNeue.ttc
HelveticaNeueDeskInterface.ttc
Menlo.ttc

Fonts updated in 10.14.3

The 10.14.3 delta update contains all of the same updated fonts as 10.14.2 (except Apple Color Emoji), but there are no changes. They are exactly the same versions and sizes as previous.



macOS, 10.13 High Sierra

I like easy. This version of the macOS turned out to be like that. The minimum fonts are almost the same as Sierra. There are quite a few more San Francisco fonts than previous. In Sierra, there were 33. In High Sierra, there are now 58.

An early update to High Sierra changed something that affected the display of emojis in Messages. Possibly a framework installed for the Safari 11.0.1 update. Whatever the cause, Messages now requires the fonts AppleSDGothicNeo.ttc and ヒラギノ角ゴシック W3.ttc to display emojis. Otherwise, all you get is the question mark in a box from the font LastResort. These two fonts have been added to the minimum font list for the System folder.

The five (now six) fonts that will not work properly in El Capitan or Sierra (Athelas.ttc, Charter.ttc, Marion.ttc, Seravek.ttc and SuperClarendon.ttc) continue to be a problem in High Sierra. The issue was momentarily fixed in Sierra, but they went missing again as of 10.12.2. These five fonts remain in limbo with High Sierra. Iowan Old Style.ttc has joined this list in High Sierra.

When I first tested the original five known problem fonts upon High Sierra's initial release, renaming the fonts still worked. One of the recent updates to High Sierra caused renaming the fonts to stop working. It made me wonder if removing the buried .ATSD and .fontinfo data still worked.

I booted into Recovery mode to turn SIP off, then removed all of the info data for those fonts. Nothing! Not after a restart, clearing font caches, or renaming the fonts on top of removing the data info. None of Apple's apps will recognize these six fonts as being on the system. Microsoft Office, Adobe's and everybody else's software does. Just not Apple's.

At this time, the only fix is to copy these fonts from Yosemite and completely replace the High Sierra versions. That is, if you need to use these six particular fonts in Pages, TextEdit, or whatever Apple software you're using.

10.13.4 updated quite a few fonts, one of which was Charter.ttc. That font now works everywhere. Yay! However, Apple continues to ignore the other five that *don't* work. Boo! You still need to pick up Athelas.ttc, Iowan Old Style.ttc, Marion.ttc, Seravek.ttc and SuperClarendon.ttc from Yosemite if you want to use them in any of Apple's supplied apps.

Here's one I didn't think I'd ever see, and is something press and prepress shops in particular will love! Though I can't fully guarantee this, it looks like Helvetica and Helvetica Neue are now completely unhooked from the OS. Removed from the System folder, there wasn't a single OS supplied app that wouldn't launch, or behave incorrectly with them gone. I would have to assume the OS and all Apple supplied apps now use San Francisco for all display purposes.

Experiment as you wish with this possibility if you're of the group that has to wrestle with the OS versions of Helvetica and Helvetica Neue conflicting with your older Type 1 PostScript versions. All of the Adobe CC 2018 apps I have installed launched without either font set active. But, Premiere Pro displayed boxed question marks (from the system font, LastResort) where the timer numbers should be. So it is obviously using either Helvetica or Helvetica Neue. Office 2016 surprised me. Given the fact it wouldn't even launch in its earlier point release versions if Helvetica Neue was missing; Word, Excel, PowerPoint and Outlook all launched and behaved as if nothing were wrong.

This is all moot for most users. Helvetica and Helvetica Neue are used rather extensively on the web. And other than people like me who have to work around these fonts in prepress, there isn't a good reason to remove them. So, I've left Helvetica and Helvetica Neue as required fonts.

Apple has just about killed off their proprietary .dfonts in favor of OpenType. There are only three .dfonts left in High Sierra. Courier, Geneva and Monaco. Even the fonts in `/Library/Application Support/Apple/Fonts/` are now all .ttf or .ttc. The only three that aren't are in the Deprecated folder.

To remove fonts from the `/System/Library/Fonts/` folder in High Sierra, [disable System Integrity Protection](#). With SIP disabled, you can put the system fonts you don't want in the trash and empty it. No need to first restart the Mac before the OS will let you do that. Re-enable SIP when you've finished removing the fonts.

Or, you can startup to another bootable drive and remove system fonts from the non-startup drive without disabling SIP at all. Only your admin credentials are required.

The minimum fonts recommended for High Sierra in the `/System/Library/Fonts/` folder are:

- Apple Color Emoji.ttc
- AppleSDGothicNeo.ttc
- Avenir.ttc
- Courier.dfont
- Geneva.dfont
- Helvetica.ttc
- HelveticaNeue.ttc
- HelveticaNeueDeskInterface.ttc
- Keyboard.ttf
- LastResort.otf
- LucidaGrande.ttc
- Menlo.ttc
- Monaco.dfont
- PingFang.ttc
- SF... (all San Fransisco fonts)
- Symbol.ttf
- Times.ttc
- ZapfDingbats.ttf
- ヒラギノ角ゴシック W3.ttc

Fonts updated in 10.13.1

- Apple Color Emoji.ttf
- Gurmukhi MN

Fonts updated in 10.13.2

In the `/System/Library/Fonts/` folder:

- Apple Color Emoji.ttc
- ArabicUIDisplay.ttc
- ArabicUIText.ttc
- Geneva.dfont
- SFNSSymbols-Regular.otf

In the `/Library/Fonts/` folder:

- DIN Condensed Bold.ttf
- Malayalam MN.ttc
- Malayalam Sangam MN.ttc
- Oriya MN.ttc
- Oriya Sangam MN.ttc

Fonts updated in 10.13.4

In the `/System/Library/Fonts/` folder:

- Apple Color Emoji.ttc
- ArabicUIDisplay.ttc

ArabicUIText.ttc
GeezaPro.ttc
SFNSDisplay-BlackItalic.otf
SFNSDisplay-BoldItalic.otf
SFNSDisplay-HeavyItalic.otf
SFNSDisplay-LightItalic.otf
SFNSDisplay-MediumItalic.otf
SFNSDisplay-RegularItalic.otf
SFNSDisplay-SemiboldItalic.otf
SFNSDisplay-ThinItalic.otf
SFNSDisplay-UltralightItalic.otf
SFNSDisplay.ttf
SFNSText.ttf
SFNSTextItalic.ttf

In the `/Library/Fonts/` folder:

Charter.ttc
DIN Alternate Bold.ttf
DIN Condensed Bold.ttf
Gujarati Sangam MN.ttc
Rockwell.ttc

Fonts in the 10.13.5 delta update

There are many fonts in this update that were supplied with 10.13.4. However, none of them are actually new. All of the fonts in the 10.13.5 update have exactly the same sizes, creation dates and version numbers as in 10.13.4.



macOS, 10.12 Sierra

Overall, it appears Apple spent a fair amount of time in Sierra cleaning up font issues and minimizing the use of fonts by both the system and their supplied apps. A nice improvement over El Capitan.

To remove fonts from the `/System/Library/Fonts/` folder in Sierra, [disable System Integrity Protection](#). A big change over El Capitan is that once you disable System Integrity Protection, you can put the system fonts you don't want in the trash and empty it. No need to first restart the Mac before the OS will let you do that. Re-enable SIP when you've finished removing the fonts.

Or, you can startup to another bootable drive and remove system fonts from the non-startup drive without disabling SIP at all. Only your admin credentials are required.

Once you've reduced the fonts on your system in Sierra to match the lists here, Font Book continues to show many Asian fonts you've removed in the All Fonts heading as grayed out entries. These are subset fonts that used to be within the Font Book application itself in a folder named Stub Fonts. They're now deep in the system at `/System/Library/Frameworks/ApplicationServices.framework/Versions/A/Frameworks/ATS.framework/Versions/A/Support/FontSubsets`.

Like the old Stub Fonts, and as the name of the folder implies, they're only subsets of the actual fonts in the `/System/Library/Fonts/` and `/Library/Fonts/` folders. Why bother with them at all? They may be there because of users like me who need to reduce the number of fonts on their system as much as possible. They may also be the reason Messages now works in Sierra without having a bunch of extra fonts open no other apps need, as it does in El Capitan.

Unlike in El Capitan, Apple Braille and ArialHB are no longer needed for the Keyboard Viewer to properly display. The check marks in Cocoa menus also display without the need for ArialHB. From what I can tell, ArialHB is no longer a required font in Sierra. I tested Xcode 7.3.1 (the current version at the time of writing) under Sierra. And again, unlike El Capitan, all UI elements displayed normally; with or without Apple Braille.

Since I haven't yet found a need for them in Sierra, I've set the list to minimize fonts as much as possible by eliminating Apple Braille and ArialHB. You can always put copies of them back if you discover a need for either.

The case of the five fonts that will not work properly in El Capitan (Athelas.ttc, Charter.ttc, Marion.ttc, Seravek.ttc and SuperClarendon.ttc) has returned. It was fixed in beta and in the initial release of Sierra, but as of 10.12.2, is back. The fix is the same as in El Capitan (see below).

Terminal uses a new San Francisco font set as the default, which is within the application package itself. If you click through the presets in Terminal's preferences, they vary between SF Mono Regular, Courier and Monaco as the default font. If you remove Courier and/or Monaco from the system, Terminal automatically fills in the default with SF Mono Regular. So if you want to trim the number of installed fonts even further, you can remove Monaco. I've left it as a required font mostly because it's been with the Mac OS since practically day one, but I haven't found any other part of the OS that uses it.

The minimum fonts recommended for Sierra in the `/System/Library/Fonts/` folder are:

- Apple Color Emoji.ttc
- Avenir.ttc
- Courier.dfont
- Geneva.dfont
- Helvetica.dfont
- HelveticaNeue.dfont
- HelveticaNeueDeskInterface.ttc
- Keyboard.ttf
- LastResort.ttf
- LucidaGrande.ttc
- Menlo.ttc
- Monaco.dfont
- PingFang.ttc
- SF... (all San Francisco fonts)
- Symbol.ttf
- Times.dfont

Fonts updated in 10.12.2

- Apple Color Emoji.ttf

Fonts updated in 10.12.4

- Apple Color Emoji.ttf
- Symbol.ttf

Fonts updated in 10.12.5

- Apple Color Emoji.ttf

Symbol.ttf is included as an updated font, but is no different from the 10.12.4 update version. The size and version number is identical. If there were any changes, Apple must have considered them too minor to even bother changing the version number.

Fonts updated in 10.12.6

- Apple Color Emoji.ttf



OS X, 10.11 El Capitan

El Capitan's main font set is now San Francisco. This is used for the top menu bar and other elements. HelveticaNeueDeskInterface.ttc is used for the menu items that drop down from the top menu choices, as is Lucida Grande for some of the special characters. All are critical to at least fully see what you're doing from the desktop. So if the San Francisco fonts are active, but HelveticaNeueDeskInterface.ttc and LucidaGrande.ttc are missing, El Capitan will show the main menu bar text and all drop down menus will be blank.

I was hoping Apple would combine all of the San Francisco fonts into an OpenType .ttc package just for the sake of tidying up the `/System/Library/Fonts/` folder after the beta period was over, but they didn't. Being PostScript OpenType fonts, they also can't be bundled together in a suitcase as TrueType OpenType fonts can. That makes the minimum font list appear really long, but isn't as many fonts as it looks like.

There are some odd ones in here because of the way Apple changed which fonts the Keyboard Viewer looks for. It no longer uses AppleSDGothicNeo. For whatever reason, it now uses Apple Braille, ArialHB (Hebrew), Lucida Grande and PingFang. If ArialHB is missing, the Keyboard Viewer will display as a completely unreadable mess. If you remove Apple Braille, it will display with one small anomaly, but will still be entirely usable.

A reader has informed me that ArialHB is used to render the check marks in Cocoa menus, such as font menu in TextEdit. Apple Braille is

used in some other UI elements, such as the Xcode toolbar display. So for at least this version of OS X, these are now fixed fonts for various, and oddly used purposes. An observant reader noted that Avenir is required for Maps. Without it, the maps won't zoom or otherwise move.

So close, but no cookie. Helvetica is now fully back in control of the user. Because you can remove Helvetica.dfont from the `/System/Library/Fonts/` folder in this version of OS X, you can use your Type 1 PostScript versions without interference from the OS. Because some Adobe apps, and possibly other third party titles won't launch if Helvetica is missing, I've left it as a required font. See section 6 for more detailed information on the Adobe apps and Helvetica. Apple has almost handed back control of Helvetica Neue. Notes is the only remaining OS X supplied program that requires HelveticaNeue.dfont to launch. However, some third party apps make it literally impossible to remove HelveticaNeue.dfont without causing them to crash.

PingFang replaces Lucida Grande where the OS needs to display Asian Kanji characters on an English system. If it's missing, such characters will display as a question mark in a box. With this minimum font list, you will at times briefly see a string of question marks in a box in Safari's URL/search bar when changing web sites. But these are always fixed back to readable text in less than a second.

Speaking of question marks in a box, be careful which fonts you remove from the `/System/Library/Fonts/` folder. If you remove the wrong one, when you perform a Boot Manager startup (restart and hold the Option key), the name of the drive with El Capitan on it will display as a string of boxed question marks. I didn't take the time to figure out exactly which removed font causes this, but I believe it's HelveticaNeueDeskInterface.ttc. Simply putting the necessary font back doesn't fix the problem. The only way to fix it is to first make sure you have all fonts listed here for El Capitan in the `/System/Library/Fonts/` folder. Then make a full backup of the drive, erase the partition El Capitan was on, and restore the backup.

There are a total of five fonts that do not work properly in El Capitan. They are:

Athelas.ttc
Charter.ttc
Marion.ttc
Seravek.ttc
SuperClarendon.ttc

A fellow by the name of Mike Detwiler is the person who informed me Super Clarendon doesn't work. We got to emailing at length about the subject. After much testing between us over two or three days, we discovered the problem is not the fonts themselves. Rather, System Integrity Protection is (mostly) to blame. Many thanks to Mike for helping to figure out what was going on.

As installed by OS X, these fonts don't appear in TextEdit at all, only partially in Mail, and so on in the Apple supplied apps. At the same time, they would function normally in the Adobe CC apps, Office 2016, Quark XPress 2015 and other third party apps.

If you simply rename the fonts on the desktop, they suddenly work in all apps! Okay, why is that? The file name of a font quite literally means nothing as to whether it will activate or not. So why did it work for these? The answer has to do files installed deep in the System folder. The OS installs font name matched .ATSD and .fontinfo files in the `/System/Library/Frameworks/ApplicationServices.framework/Versions/A/Frameworks/ATS.framework/Versions/A/Resources/FontInfo` folder for every font OS X installs. Such as this for Marion.ttc:

Marion.ttc.DF798_0.ATSD
Marion.ttc.DF798_0.fontinfo

They've been there as part of the Mac OS for a long time, but until now, never really had any logical reason to exist. At least not to me. Since this issue didn't exist before El Capitan, they now appear to be tied to System Integrity Protection. We believe this is what's happening.

- 1) The .ATSD and .fontinfo files haven't been correctly updated by Apple for these five fonts. As such, it considers those fonts as having been altered. Being altered, SIP assumes possible malware injection and refuses to use them.
- 2) When you rename those fonts, SIP no longer has a file name that matches the protection scheme and considers them third party fonts.
- 3) The only part of this that is still somewhat off base is that these five fonts still don't work even if you have SIP disabled, and renaming them still fixes the problem.
- 4) When you replace these first four fonts with the Yosemite versions, then the expected data must match the .ATSD and .fontinfo files, as they then work without renaming the fonts. Super Clarendon remains unrepentant no matter what version of OS X you copy it from.

You can prove the issue is tied to these framework files by moving the .ATSD and .fontinfo files for those five fonts to a new folder on the drive. Now enable the normally named El Capitan fonts in any font manager. With no reference to them in the framework file, the OS considers them third party fonts (or something) and they work.

There are three ways to get around this. The first two require first temporarily disabling System Integrity Protection (see just below how to do this).

- 1) Rename the fonts. Like Marion.ttc to MarionE.ttc. It literally doesn't matter what you change the file name to.
- 2) Leave the file names as they are and remove their matching .ATSD and .fontinfo files from the framework folder.
- 3) A few more steps, but this still takes less time than disabling SIP to perform 1 or 2, and then re-enabling SIP.

1. Highlight each of the five fonts and press Command+D to duplicate them. For example, this will produce an "Athelas copy.ttc" file.
2. Highlight each *original* font and choose Get Info from the menu bar, or press Command+I. Change the permissions of each to add yourself (the admin account user) with Read/Write privileges.
3. Move the original, normally named fonts to the trash and restart. Empty the trash.

These same fonts also didn't work in the Sierra beta until just recently. The beta 3 update has fixed the issue, so you can expect the fix to carry through to the final release. Whether or not we'll see another point update for El Capitan to fix it in this version of OS X remains to be seen. But at least you have these three methods to use if not.

There is an activation/deactivation issue in El Capitan that exists in at least 10.11.4 through 10.11.6. This affects all font managers, including Apple's Font Book. What happens is you activate some fonts and use them in a document. Deactivate them and any text you were working on in TextEdit, MS Word and other apps will do what you would expect; the text reverts to an available font. Enable the fonts again, then disable them. Almost without fail the second time, the fonts you just disabled will continue to show in your app. You can even click on a line and continue typing in that font as if it's still available. Extensis also tested this when the fonts appear to remain, and the system shows that yes, the fonts have been deactivated. What seems to be happening is that El Capitan is continuing to work from a persistent cache. If you close the document and reopen it, *then* it will show the fonts as missing again. The only method that works 100% of the time to avoid this visual confusion is to manually place fonts in the Fonts folder of your user account, and take them out when you're done using them.

Okay, the main El Capitan question. How do you remove fonts from the System folder? The new System Integrity Protection won't let you remove anything via the desktop as an admin user, or even in Terminal using sudo. There are only two ways around this. The first is the preferred method.

- 1) Boot to any other partition or drive you have macOS on. Even another volume with El Capitan on it. Once there, any other drive is just a drive full of files. You can now remove fonts from the `/System/Library/Fonts/` folder of the non startup drive. You do need to provide your admin password to complete the action. You must delete the fonts from the El Capitan drive you are modifying while you are booted to another volume. If you don't and boot back into the El Capitan drive you were changing, it will insist the system fonts you removed are still in use, even if they're in the trash. Once all unnecessary fonts have been deleted, restart again to your main drive.

If you have no other physical drive to boot to, you should. All drives die, including SSD drives. Having your valuable data in only one place is a sure way to eventually lose it. In the case of a drive that dies, you then also have another bootable drive to use immediately. Purchase an external drive and then partition it as GUID and format it as Mac OS Extended (Journaled). You can then use Disk Utility, or third party apps such as [SuperDuper!](#) or [Carbon Copy Cloner](#) to clone your main drive to your external drive. Then you can startup to that drive and follow these directions.

- 2) If you would rather continue on with method two, first [disable System Integrity Protection](#).

You can now remove fonts from the `/System/Library/Fonts/` folder while booted to El Capitan. The OS will behave the same as in Yosemite and earlier. Put the fonts you want to remove in the trash and enter your admin password when prompted. Do not re-enable System Integrity Protection yet.

You must empty the trash before re-enabling System Integrity Protection. You can't empty the trash yet since you moved active system fonts to the trash during that same login session, and you can't delete them from the trash with SIP enabled. Restart and then empty the trash while the system fonts you removed are now considered inactive. Once you have pared down your fonts in the `/System/Library/Fonts/` folder and deleted them, startup into Recovery Mode again and re-enable System Integrity Protection.

A user found that some Emojis would work in Messages, but many others would display as simple bars. Turns out, like the Keyboard Viewer, Messages is very strangely tied to system fonts that have no logical reason to be used for this app in order for it to work properly. All of the following fonts are required for Messages. If any one of them is missing, Emoji use in Messages breaks again. If you don't use Messages, or never use Emojis in that app (and don't care if you see the ones sent by others), you can skip them. All are in the `/System/Library/Fonts/` folder.

AppleSDGothicNeo.ttc
AquaKana.ttc
Hiragino Sans GB W3.ttc
Hiragino Sans GB W6.ttc
ヒラギノ角ゴシック W0.ttc
ヒラギノ角ゴシック W1.ttc

ヒラギノ角ゴシック W2.ttc
ヒラギノ角ゴシック W3.ttc
ヒラギノ角ゴシック W4.ttc
ヒラギノ角ゴシック W5.ttc
ヒラギノ角ゴシック W6.ttc
ヒラギノ角ゴシック W7.ttc
ヒラギノ角ゴシック W8.ttc
ヒラギノ角ゴシック W9.ttc

The minimum fonts recommended for El Capitan in the `/System/Library/Fonts/` folder are:

Apple Braille.ttf
Apple Color Emoji.ttf
ArialHB.ttc
Avenir.ttc
Courier.dfont
Geneva.dfont
Helvetica.dfont
HelveticaNeue.dfont (for Notes)
HelveticaNeueDeskInterface.ttc
Keyboard.ttf
LastResort.ttf
LucidaGrande.ttc
Menlo.ttc
Monaco.dfont
PingFang.ttf
SF... (all San Fransisco fonts)
Symbol.ttf
Times.dfont

Fonts updated in 10.11.1

Apple Color Emoji.ttf

Fonts updated in 10.11.2

I would normally list all font revisions by name, but there are so many in 10.11.2, I'm going to summarize. In the System folder, Apple Color Emoji.ttf, PingFang.ttf, 24 of the 33 SNFSText... fonts, and 5 of the 12 Hiragino Sans fonts have been updated. In the root Library folder, YuMincho.ttf and ヒラギノ丸ゴ ProN W4.ttc have been updated.

Fonts updated in 10.11.4

Almost all of the same fonts updated in 10.11.2 are included in the 10.11.4 update. However, only Apple Color Emoji.ttf is actually any different. All others are 100% identical to those in the 10.11.2 update, except for the creation date.

Required fonts in the `/Library/Fonts/` folder.

The `/Library/Fonts/` folder contains many standard web fonts. At minimum, the following fonts should remain in order for web pages to display properly. The names below are how they will appear in El Capitan, 10.11 through Mojave, 10.14.

Recommended minimum fonts for the `/Library/Fonts/` folder:

AppleGothic.ttf
Arial.ttf
Arial Bold.ttf
Arial Italic.ttf
Arial Bold Italic.ttf
Arial Black.ttf
Arial Narrow.ttf
Arial Narrow Bold.ttf
Arial Narrow Italic.ttf
Arial Narrow Bold Italic.ttf
Comic Sans MS.ttf

Comic Sans MS Bold.ttf
Georgia.ttf
Georgia Italic.ttf
Georgia Bold.ttf
Georgia Bold Italic.ttf
Impact.ttf
Tahoma.ttf
Tahoma Bold.ttf
Times New Roman.ttf
Times New Roman Bold.ttf
Times New Roman Italic.ttf
Times New Roman Bold Italic.ttf
Trebuchet MS.ttf
Trebuchet MS Bold.ttf
Trebuchet MS Italic.ttf
Trebuchet MS Bold Italic.ttf
Verdana.ttf
Verdana Bold.ttf
Verdana Italic.ttf
Verdana Bold Italic.ttf
Wingdings.ttf
Wingdings 2.ttf
Wingdings 3.ttf
Webdings.ttf

All other fonts in the [/Library/Fonts/](#) folder can be removed.

If you use iLife or iWork: The following fonts are also located in the [/Library/Fonts/](#) folder. While the iLife and iWork apps will launch without the following fonts, the supplied templates use them. Apple suggests these fonts always be available for these apps. If you do not use any of the iLife or iWork apps, you can remove these fonts.

American Typewriter.ttc
Baskerville.ttc
BigCaslon.ttf
Chalkboard.ttc
Cochin.ttc
Copperplate.ttc
Didot.ttc
Futura.ttc
GillSans.ttc
Herculanum.ttf
Hoefler Text.ttc
Optima.ttc
Papyrus.ttc

Of the above list, the font Optima.ttc is in the [/System/Library/Fonts/](#) folder.

If you use Microsoft Office: See Section 4 of this article for more info on the fonts Office installs. Some are newer and some are older than those installed by Leopard through Yosemite.

Once you have manually reduced the fonts on your system to the minimum, always use your font manager to control all other font activation and deactivation. The better font managers will stop you from creating font conflicts. Your font manager can't help prevent that if you manually place fonts you want to use in a Fonts folder.

An important step you should take after manually removing fonts is to clear the font cache files from the system. See section 17 for the proper procedure. Font Book users should also reset the application to update its database. See section 7 for more details.

When it comes to font managers, there's one thing I can't stress enough: **have only one font manager on your Mac at a time.**

When you double click a font, macOS no longer launches multiple font managers if they're installed, and hasn't for a while. It will only launch the one assigned to your fonts. Though even that can mean different font managers if for instance, Font Book is still assigned to older legacy Mac TrueType fonts, and everything else to your third party font manager. That's a problem when you don't really want to use Font Book. Having Font Book's database on the system can prevent another font manager from working correctly, and just the act of having

Font Book launch creates a new database. Then you have to remove the database again. So if you aren't using it, you should not have Font Book on the drive. See section 7 on the steps for completely removing Font Book and its database.

A scenario of having more than one active font manager: You open a font in Suitcase. Then later, you open the same font in FontExplorer X Pro. You then disable that font in Suitcase. However, the font is still active in all of your applications. Why? Because FontExplorer X Pro is still holding the font open. I used Suitcase and FontExplorer X Pro for this example, but this will happen in virtually any case of multiple font managers on your system. Once you have decided which font manager you are going to use, completely remove any other font manager from your Mac.

Enabling and disabling System Integrity Protection

To enable or disable SIP, you can:

- 1) Restart and hold down the Command+R keys to boot into Recovery Mode.
- 2) Restart and hold down the Command+Option+R keys to boot into Internet Recovery Mode.
- 3) Start up to a bootable USB flash drive you have created using a full installer of El Capitan or later. Use the OS version that matches what you have installed on your Mac. Plug in the flash drive, restart the Mac and immediately hold down the Option key. Choose your flash drive from the menu of available startup drives with either the mouse or keyboard.

Once at the main work screen, launch Terminal from the Utilities menu. Enter the command:

```
csrutil disable
```

You should get a message that System Integrity Protection has been successfully disabled. Then enter the command:

```
reboot
```

You can now remove fonts from the `/System/Library/Fonts/` folder while booted to Sierra or later. The OS will behave the same as in Yosemite and earlier. Put the fonts you want to remove in the trash, enter your admin password when prompted, and then empty the trash. El Capitan is different. It won't let you empty the trash until you restart the Mac. Only after you've emptied the trash can you re-enable SIP.

As of High Sierra 10.13.4, you can no longer remove any OS installed app that are on the startup drive using the Finder. With SIP enabled, the OS tells you the app is required by the OS. With SIP disabled, you used to be able to remove any apps by placing them in the trash and entering your admin password. You now get a slightly different message saying the item can't be deleted.

This leaves two options:

- 1) Boot to another drive or partition so you can delete whatever you want from the non-startup drive. Only your admin password is required to complete the action.**
- 2) With SIP disabled, use Terminal to delete unwanted apps. Mainly, Font Book for the purpose of this article. The command to remove Font Book is noted in section 7.**

With SIP disabled, this change does not affect the removal fonts in the System folder. Only your admin password is required.

It's not a good idea to leave System Integrity Protection disabled. The idea of it is to protect system folders from malware that already exists (mostly in the form of adware at this time), and is beginning to become more prominent in macOS. With SIP enabled, such software will not be able to install to critical system folders. If they do (such as kernel extensions), the software will not be allowed to run. SIP also protects apps installed by macOS from injection of unwanted code.

Once you have pared down your fonts in the `/System/Library/Fonts/` folder and deleted them, startup according to one of three above methods again. Only this time in Terminal, enter:

```
csrutil enable
```

You should get a message that System Integrity Protection has been successfully enabled. Then enter the command:

```
reboot
```

Known issues with disabled fonts

This section describes other fonts you may need available at all times depending on the software you use.

Geneva

I have found no reference that requires Geneva, but it's best to leave it as it has long been a standard font for the Macintosh OS.

Helvetica

Some of the macOS supplied applications that will not launch if Helvetica is missing are mentioned in Apple Knowledge Base article [TA22311](#). While not listed in that particular article, iChat is another application which requires Helvetica, as does iCal. If you've seen the message, "Internal Error" when starting iChat, you're missing Helvetica.

Some Adobe CS/CC apps will not launch if Helvetica is missing.

Mail in High Sierra still has a dependency on one of the Helvetica fonts. It will launch and display normally if Helvetica is missing, but if you do a Search, the Results listing will fall back to LastResort.

Helvetica Neue

The Office 2016 apps will not launch if Helvetica Neue is missing (earlier point releases only, now fixed).

There's no good reason to remove the macOS supplied versions of Helvetica or Helvetica Neue unless you work in prepress, advertising, design, etc., where you must be able to use a different version of these type faces (see section 6). If you don't, leave them be.

iPhoto 7 and iLife 08 both require the presence of Helvetica Neue.

Mail also requires MarkerFelt, or the Notes portion of the application won't open. Notes has been separated as an individual app from Mail starting in Mountain Lion 10.8.

Times

If you are using the Monaco (X-Rite) GamutWorks application, it requires Times to function. This can be the Times.dfont supplied with macOS, an OpenType or a Type 1 PostScript version. But it must be Times. Times New Roman or any other variation of the Times typeface will not work. This is actually outdated information since GamutWorks is PowerPC only software.

Apple's Grapher application also requires Times. Like Monaco's GamutWorks, it must specifically be Times. In addition, Grapher also requires Symbol. Without Times, Grapher will not launch, instead giving you a message that there was a problem with the application. If you have Times open but not Symbol, Grapher will launch but then erroneously tell you there is a font conflict, when in reality it's just that Symbol is missing.

Symbol

MailMate requires Symbol.

Zapf Dingbats

iTerm2 requires Zapf Dingbats.

ヒラギノ角ゴ ProN W3

If you are using DxO Optics Pro, even if English is set as your language, it will fail to launch if this font is missing.

Apple Gothic

In macOS, the Messages app will not launch if Apple Gothic is missing.

Fonts installed by Microsoft Office

El Capitan, 10.11 through Mojave, 10.14 include many of the same named fonts Office 2011 installs on your system.

Any time you manually remove fonts, you should then clear the font cache files from the system (section 17) and reset Font Book's database (section 7).

Office 2011:

The fonts installed by Office 2011 go to the [/Library/Fonts/Microsoft/](#) folder. Therefore, it creates conflicts with the existing similar fonts installed by macOS.

The Office 2011 installation creates a mixed bag of older and newer conflicting fonts. You need to manually sort them out as described next.

During the initial install of Office 2011, newer .ttf font sets replace those installed by macOS. Though some in El Capitan through Mojave are newer. The OS installed fonts are deprecated to a new folder named [/Library/Fonts Disabled/](#). They are:

- Arial.ttf
- Arial Italic.ttf
- Arial Bold.ttf
- Arial Bold Italic.ttf
- Brush Script.ttf
- Times New Roman.ttf
- Times New Roman Italic.ttf
- Times New Roman Bold.ttf
- Times New Roman Bold Italic.ttf
- Verdana.ttf
- Verdana Italic.ttf
- Verdana Bold.ttf
- Verdana Bold Italic.ttf
- Wingdings.ttf
- Wingdings 2.ttf
- Wingdings 3.ttf

There is no need to keep the fonts in the [/Library/Fonts Disabled/](#) folder since identically named versions have been installed. You can delete the entire **Fonts Disabled** folder if you wish. If you want to keep the latest revisions, you'll have to do a Get Info on each of these fonts in both the [/Library/Fonts/](#) and [/Library/Fonts Disabled/](#) folders. Then check the version numbers and toss the older version of each. Any newer versions still in the **Fonts Disabled** folder then need to be moved back to the [/Library/Fonts/](#) folder.

Office 2011 also installs conflicting fonts which are **older** than those supplied with El Capitan, 10.11 through Mojave 10.14. These are Mac legacy TrueType suitcase fonts, as opposed to much newer .ttf versions supplied with El Capitan through Mojave. So keep the .ttf fonts supplied with macOS and remove the following Mac legacy TrueType suitcase fonts from the [/Library/Fonts/Microsoft/](#) folder:

- Andale Mono
- Arial Black
- Arial Narrow
- Arial Rounded Bold
- Comic Sans MS
- Georgia
- Impact
- Tahoma
- Trebuchet MS

For at least the last couple versions of Office, Microsoft has used the font family Cambria as the default for Word and Outlook. If you prefer to use some other font as your defaults, you can change it. In Word 2011, open the Normal.dotm template in your user account at [~/Library/Application Support/Microsoft/Office/User Templates/](#). Press Command+D to call up the font palette. Choose the font and point size you prefer, then press the Default button at the lower left. Click OK. Close the template and save. All new documents will now open with your chosen default font rather than Cambria. For Outlook, open the preferences and click on the Fonts icon. Change the options to your preferred font and close the preferences. You must do this before removing any fonts Outlook uses by default. If either font in its preferences are not available when you click on the Fonts icon, the preference won't open.

Once you have your preferences set for Word and Outlook, you can remove the rest of the fonts Office 2011 installs. If you use any of the preset documents from the Document Gallery, they will still open even if you remove every font Office installs. Any missing font the template looks for will simply be substituted with an existing font. Clicking anywhere on the substituted text will show the font name Word is looking for in the font selection drop down menu.

Office 2016:

Information from Microsoft is being noted here. As users have likely already realized for themselves, when you dig into an application package and modify anything, you are taking the risk of causing the program to function incorrectly. In all cases of such modifications, as relayed by a Microsoft engineer, doing so "breaks the code signature of the app, and is unsupported." So, proceed at your own risk and keep that in mind for any of the following information for Office 2016.

The early issue of the Office 2016 apps refusing to even launch if Apple's Helvetica Neue was missing from the System folder has been fixed. Though you must be using one of the point releases close to 15.38 or higher. I didn't notice exactly when that dependence on Helvetica Neue was resolved, but I do know it was fairly recent.

Much of what used to be in this section was obsolete, as long as you have Office 2016 up-to-date. And you should anyway since there have been many bug fixes and feature improvements since its initial release. Because of that, I've removed anything having to do with point releases earlier than 15.22.

Office 2016 no longer installs any fonts to the `/Library/Fonts/` folder. All fonts installed by Office 2016 are embedded within each application package. They are within a folder named `Fonts`. All fonts in the `DFonts` folder are extras that do not affect Office 2016's ability to function, and that entire folder can be removed.

In El Capitan, legacy Mac TrueType and Type 1 PostScript fonts not working in Office 2016 is not Microsoft's fault. It's El Capitan's. It's the same issue as described in section 6. If you are seeing this issue, this bug in El Capitan has been fixed as of 10.11.3. Update the OS as necessary.

15.22 through 15.41

Within each application package, the folder `FontDigestWithNoSystemFonts` has appeared. This folder and its enclosed items exist to automatically prevent the Office 2016 apps from displaying the names of Mac OS supplied fonts you've removed from the drive. So there's no need to modify or remove any `.plist` files in the Office 2016 application packages to fix this particular issue.

That's good, but now the opposite problem has appeared. If you remove the embedded `DFonts` folder in each app to clear out the large number of supplied fonts you don't want on the system, their names continue to display in all Office 2016 apps as if they still exist. Attempting to remove all `.plist` files as you could in earlier point releases causes the Office apps to show no fonts at all in its lists, other than Cambria.

Right click on an Office 2016 app and choose `Show Package Contents`. Within that, go to the `Contents/Resources` folder.

Delete the `DFonts` folder. In list view (Command+2), click on `fontFamilyImages1025.plist` to highlight it. Scroll down to `fontsImages2052.plist` and Shift+click on it to select that file and everything in between. Scroll back up and Command+click on the folder `Fonts` to deselect that one item. Delete the highlighted `.plist` files.

Repeat for the remaining Office 2016 apps.

Now for the fun part. You need to deal with the `fontFamilyImages.plist` and `fontsInfo.plist` files to fix the font lists so Office 2016 doesn't keep displaying the names of fonts you just removed by deleting the `DFonts` folder. Fortunately, this turned out to be much easier than I expected.

First, download `BEdit`. You need a text editor that can properly display, edit and save an XML based `.plist` file. Open `fontFamilyImages.plist` and `fontsInfo.plist` with `TextWrangler` and edit them so that both appear as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict/>
</plist>
```

You can remove most of the large amount of content quickly by clicking somewhere underneath the text, `<plist version="1.0">`. Scroll all the way down to the bottom of the `.plist` data and Shift+click somewhere near the bottom. That will highlight everything in between and you can delete the entire block of highlighted data in one move. Complete any necessary editing to duplicate the image shown here. `TextWrangler` may need your admin password to complete the edit of these files. Enter that when asked so it can save your changes.

You would think removing essentially all font related data from these `.plist` files would cause the same problem as simply deleting them, but it doesn't. When you launch the Office 2016 app you edited these two `.plist` files in, only fonts that exist on your Mac and what you've left in the `Fonts` folder of the Office 2016 application package will appear in the font lists. You also only need to do this `.plist` editing once. For the

rest of the apps, you can copy your edited files to the each of the remaining apps, replacing the existing items.

At this point, you're done. If you want, you can also clear most items out of the Fonts folder in each Office 2016 application package. A lot of them duplicate fonts that are already on the system. If you've gone by the required font lists in this article, Tahoma, Wingdings, Webdings and others already exist as OS X supplied fonts. There's no need for them to be here, too. Personally, I remove everything but the Calibri and Cambria sets. For Outlook, you must keep its version of symbol.ttf within the application package, or it won't display regular bullets in your emails. The safest course though is to leave the embedded Fonts folders intact and let Office sort out what it does, and doesn't use.

A very big note I hadn't thought of. All of the Office 2016 instructions above refer to the perpetual license, stand-alone version. They do not work properly with the volume licensed version of Office 2016. Much, much thanks to a reader for the following information as I only have the single user license to test with. The idea was to do the modifications above on one Mac, then copy the font fixed versions to all other Macs on the volume license so you only have to do the work once. Here's what he discovered. His text was well written, and is presented pretty much verbatim.

Making copies of the updated packages is all well and good, but it only works if you have Office 2016 preinstalled on the Macs ahead of time (for non Office365 users). If your Office 2016 suite is serialized or VL (vol license) and Office365 sign-on is NOT used, copying over updated packages before a clean install will cause an "Offline" error at startup. And for serialized VL users, it wont work to use the included "Serializer" package, even after a restart. You have to delete the updated packages and install your VL/serialized package first. The version installed doesn't matter as long as it's the same version or LOWER than the custom/modified packages created.

Once the initial install of Office 2016 is complete, then the packages can be copied over successfully. In addition, it would be recommended to open each app to complete the first-run config (welcome screen with options to login with 365 if used and/or clicking the "Get Started" prompt.

We had also both discovered a bug in Office 2016. You can't open multiple Office docs if the app you need isn't already running. If you do, it will never open and become unresponsive. You either have to launch the Office 2016 app by itself, or by double clicking only one document. Once it's running, you can open any number of items at once as you want. **This bug has been fixed as of version 16.11.1.**

16.9 through 16.16.2

Microsoft drastically changed the internal workings with the 16.x release. Because Office 2016 now follows the sandboxing rules of the App Store, it's been extremely difficult to remove fonts and have the apps behave accordingly.

This is a huge update. It includes not just a lot of security updates and fixes, but also new features and many improvements. For all intents and purposes, this amounts to a major 2018 upgrade from version 15.x to 16.x while still carrying the name, Office 2016.

A big, big thanks to MikeMuc at the German group [MacTechNews](#) for figuring out where Office 2016 hides the perpetual list of fonts, even if you've removed them. At least for now, we can once again get the number of fonts in Office 2016 down to only what's needed, and have the font lists reflect that.

Excel needs to be treated separately from Word, PowerPoint and Outlook. Also, how Office 2016 sees fonts has changed a lot since the 16.9 release. Much of the information that used to be here no longer applied, so it has been rewritten for the most recent versions.

Earlier versions of Office 2016 worked as expected, but beginning a few points releases back, Word, PowerPoint and Outlook will not list Zapf Dingbats.

Version 16.13 and forward has a new font set, TH SarabunPSK. It's in the previously empty DFonts folder of Excel. This new font set is also in Word, PowerPoint and Outlook, but is in the embedded Fonts folder.

Fixing the font list in Excel is really, really simple. The DFonts folder is still there and contains the new font set, TH SarabunPSK. You can remove that set, but Excel will continue to show it in its font list whether the font is actually there or not. Easiest solution is to just leave it. Otherwise, all you have to do for Excel is open the embedded Fonts folder and get rid of everything except the full sets of Arial, Cambria and Calibri (it won't launch without one of the Arial fonts). Open Excel and only the three remaining sets of embedded fonts, TH SarabunPSK, and what you have left on your Mac show in the font list.

A new, important note for Excel: As of version 16.21, in the DFonts folder, Microsoft has added a version of Times. If you remove it, then each time you quit Excel, you'll get an OS message that the app has quit unexpectedly. Meaning, of course, you must leave an unwanted version of Times in the application bundle. Fortunately, this doesn't really affect much regarding print production work.

Word, PowerPoint and Outlook pay no attention to the fact you already have Tahoma, Wingdings, Webdings and other fonts already on your Mac. Removed from within the application packages, they won't even appear in the font list. In other words, you have to leave those fonts embedded in order to see and use them.

The fix for these three apps is also very simple. It involves one file, and most of the fonts. For Word, PowerPoint and Outlook, right click on each, choose [Show Package Contents](#) and go to the [Contents/Resources](#) folder.

Rename, or remove the file:

applicationfontmetadata.json

If you prefer to leave the file and rename it, you can change it to anything so it can't be "found". Adding any character to the beginning of the file name will do it.

To see, and be able to use all fonts you expect in Word, PowerPoint and Outlook, leave this set in each of the application package [Contents/Resources/Fonts](#) folder. The font seguisym.ttf (Segoe UI Symbol) is necessary in order to see the right pointing arrow for a tab character, and possibly others, if you are viewing non printing (¶) characters. All other fonts can be removed.

arial.ttf
arialbd.ttf
arialbi.ttf
ariali.ttf
ArialNarrowBoldItalic.ttf
ArialNarrowItalic.ttf
ArialRoundedMTBold.ttf
ariblk.ttf
Calibri.ttf
Calibribd.ttf
Calibrii.ttf
calibril.ttf
calibrili.ttf
Calibriz.ttf
Cambria.ttc
Cambriabd.ttf
Cambriai.ttf
Cambriaz.ttf
seguisym.ttf
symbol.ttf
tahoma.ttf
tahomabd.ttf
Verdana Bold Italic.ttf
Verdana Bold.ttf
Verdana Italic.ttf
Verdana.ttf
webdings.ttf
Wingdings 2.ttf
Wingdings 3.ttf
Wingdings.ttf

Unlike Excel, you can remove the TH SarabunPSK set without it refusing to vacate the font list.

You don't really have to have any font set other than Arial in each app so they launch. But you will have problems if particular fonts are missing. For instance, without Symbol and the dingbat fonts, bullets that depend on those fonts will not display since as far as Word, PowerPoint and Outlook are concerned, they don't exist, even though the exact same fonts have been installed by the OS and are in the same place every other app you use can access them from.

You can remove Calibri and Cambria from all of the Office apps, but be aware many folks will send you files that use them. They are also, of course, the default fonts for Office. You can't even change some preferences if they're missing. It's only two sets. I'd suggest leaving them.

Office 365:

We have switched to Office 365 for use on our Macs. While I won't say never, it's unlikely I'll be adding information for further perpetual license versions of Office.

The \$99 per year fee for six Macs is a far better value than getting individual licenses for Office 2019. When you figure a perpetual license release is going to happen roughly every three years, \$300 is much, *much* less expensive. If you were to purchase six Home version editions

of Office 2019 (the one *without* Outlook) at \$150 per license, it would be \$900. For the Home and Business edition *with* Outlook, it would be a whopping \$1,500! Then you'd spend that again about three years from now if you always get the latest release when they become available. Even installing Office 365 for only two Macs is cheaper than the perpetual license Business edition.

The one time cost perpetual license is now really only beneficial to a single user who will keep using that version until it no longer works in whatever is the latest OS.

There are also these differences:

1. The code for Office 2019 was frozen back in May, 2018, so Microsoft could start changing its code and testing for a perpetual license. It has none of the features added between then and now for Office 365.
2. Office 365 will get every new feature as they become available. Not really important for the person whose documents tend to use the basic features. But for a business, it's very important that you are always able to open client documents.

Other than the activation difference - and even that is really the same procedure, except you start with a different first time installer - the app structure is virtually identical. Meaning, you can follow the same instructions above for Office 2016, and also likely Office 2019, to greatly reduce the number of fonts in the font lists.

There is one major annoyance in Office 365. The font list shows a huge assortment of cloud fonts. You can click on any of them to use one. When you do, that font is downloaded to the `/Users/your_user_account/Library/Group Containers/UBF8T346G9.Office/FontCache/4/CloudFonts/` folder, and the cloud icon goes away.

A reader by the name of Ben has discovered how to clear the cloud fonts from Office 2019. This also works for Office 365. Here are his instructions.

The list of cloud fonts is stored in a JSON file deep within your `~/Library` folder. You can't simply delete it; the next time you open an Office app, it will be regenerated. However, you can rewrite its contents so that none of the cloud fonts appear in your font lists. Here's how:

Quit all the Microsoft Office apps that you have running.

1. In the Finder menu, click "Go" > "Go to Folder..."
2. A popup with a text field appears. Paste this path into the text field, then click the "Go" button.

`~/Library/Group Containers/UBF8T346G9.Office/FontCache/4/Catalog/`

3. In the Finder window that opens, control+click (or right click) the file named "ListAll_hier.Json", then select "Open with" > "TextEdit.app". (My note. I used BBEdit, but either will work).

4. In the TextEdit menu, click "Edit" > "Select All" and press delete. Then copy and paste the following text into the now-empty file.

```
{"MajorVersion":4,"MinorVersion":5,"Expiration":14,"Fonts":{}}
```

5. Save and close the file.

Note: ListAll_hier.Json is shared by all the Office for Mac apps, so these instructions will also remove the cloud fonts from the font lists in Excel, PowerPoint, Outlook, and so on.

Personal notes on this subject: I don't need to, or want to see long lists of fonts. Most people don't realize how hundreds of active fonts (that many people also never use) slow the entire system down. They take up space in RAM. It takes longer for the computer to start up. Longer for apps to load. Longer for those apps to display font lists, etc.

More importantly is they seriously get in the way of production work. Unnecessarily long font lists are bad enough, but if I need to use a client's PostScript version of an Office embedded font, how do I know I'm using it? The lists in Office don't change. Are you looking at the font you're required to use for the project, or the one installed within the Office app? There's no easy way to tell. The owners of print shops get very cranky when a client notices their slightly modified version of (example) Bookman Old Style wasn't used, and you have to toss thirty thousand dollars worth of paper alone out the door and do the entire press run over. It's either that, or sell them what you ran - if they'll accept it - at a rather steep discount. If that sounds high to some folks, going through that much paper on a web press is *very* easy. It's a big reason these instructions exist.

By far, one of the best things Microsoft could do for users would be to allow you to *easily* remove fonts you don't need.

For more detailed information on the various Mac versions of Microsoft Office, visit [Diane Ross'](#) web site.

What to do with Suitcase

Suitcase Fusion:

All information for older versions of Suitcase can be found in the old article link. This information refers to Suitcase Fusion 8.

To manually reset Suitcase Fusion, go to the `~/Library/Extensis/Suitcase Fusion/` folder, remove the file `Suitcase Fusion.fontvault` and restart your Mac.

You can also replace this file in Suitcase's preferences. Click "New Vault" and choose the same location, which will appear by default. Say yes to replace the current vault.

Preserving fonts in the vault for Suitcase Fusion:

A warning with the above method for resetting all versions of Suitcase Fusion. If you have Suitcase set to store fonts you activate in its vault and you delete its database, they will all disappear with it. If you always activate fonts in place and never use the vault, then removing the database is safe to do at any time with any version of Suitcase Fusion.

If you are using the vault and need to reset any version of Suitcase Fusion, follow these steps first to save the fonts stored in the vault.

- 1) Your sets will be listed in the left pane under the "Font Library" heading. Highlight the first set and then Shift+click on the last set to select all. Press Command+D. It will ask you where you want to save your fonts. Navigate to an existing folder or create a new one. Choose your target folder and press the Choose button. Suitcase will save full copies of the vault fonts to that folder. They will also be saved in subfolders by the same name of all sets you had them separated by.
- 2) Quit Suitcase.
- 3) Go to your user account and remove the vault database noted above. Relaunch Suitcase Fusion.
- 4) Go to the folder you had saved your vault fonts to and drag and drop the subfolders of fonts into the Suitcase application. This will add them back to the vault and recreate your sets by the same names you were using before.

Controlling Helvetica fonts for prepress

If you work in a professional prepress environment, then this section will be important to you.

Do you need to remove Apple's Helvetica fonts?

I note this since not everyone reading this section may be clear on why you would want or need to do this. Apple made the unfortunate decision to give their Helvetica fonts the exact same internal names as the long standing Type 1 PostScript fonts from Adobe and Linotype. This creates a fixed font conflict that can be definitively resolved only by removing Apple's fonts from the system. This **does not** affect the newer OpenType Helvetica fonts available from Adobe and Linotype as they do not have the same internal names, and can therefore peacefully coexist with Apple's versions. Since it can be imperative to use the exact same version of a font that was used to create a project, the conflict with the older Type 1 Helvetica fonts makes it necessary to remove Apple's fonts.

This all started long before OS X/macOS. Apple's Helvetica fonts have always conflicted with the Type 1 versions; but back in OS 9 and earlier, no one cared. Apple wasn't using their Helvetica versions in the interface or any of its applications, so it wasn't a problem to remove Apple's Helvetica fonts from the hard drive and put your Type 1 fonts in their place. Then came OS X and Apple decided to start using Helvetica rather extensively in its applications design, forcing it to become a required font.

For the typical home user, there is no need to replace the supplied .dfont, .ttf or .ttc versions of Courier, Helvetica or Helvetica Neue. The information in this section is intended for advanced users. If you have no need to use older Type 1 PostScript or other older conflicting versions of Helvetica, Helvetica Neue or Courier, then please disregard these instructions. This avoids the possibility of your Mac becoming unusable by accidentally removing critical system fonts.

For El Capitan, Helvetica is only used for Notes, but is required for other third party apps. Helvetica Neue can be removed.

In all following cases, if you use Font Book and have manually removed Apple's Helvetica fonts, you must also reset Font Book's database. A damaged or orphaned Font Book database can also cause other font managers to be unable to activate or

deactivate fonts, even if the Font Book application is not on the hard drive. It can even prevent fonts from activating that you manually place in a Fonts folder. So whether you use Font Book or not, you must delete/reset its database after manually removing fonts from the system. See section 7 under "Resetting Font Book's database" for more details. If you have already removed Font Book and reset its database, you do not need to repeat these steps.

Removing Helvetica fonts in El Capitan through Mojave.

Copy Helvetica.dfont and HelveticaNeue.dfont to another location (these are .ttc fonts in High Sierra) and then remove them from the `/System/Library/Fonts/` folder. You will need to [disable System Integrity Protection](#), or start up to another bootable drive/partition.

Suitcase Fusion and Helvetica fonts.

Suitcase can automatically circumvent the Apple Helvetica fonts.

- 1) You can then delete the Font Book application if you wish (and actually, should).
- 2) Clear all font cache files and restart. See section 17 for more on font cache removal.
- 3) In Suitcase Fusion's preferences, make sure the radio button for "Activate the requested font" is on. Also, have the check box on for "Notify if a conflict occurs". You should always know when a font you're activating is turning another one off that you may need for a current project (this check box has disappeared in Suitcase Fusion 8).

Now you can leave Apple's Helvetica fonts right where they are in the System folder. When you open a conflicting Helvetica font (normally a Type 1 version) it will pop up a box that fonts you are opening conflict with the System fonts. Continue and Suitcase Fusion automatically deactivates the system fonts in favor of the Helvetica fonts you just turned on. When you deactivate your Helvetica fonts, the Apple Helvetica system fonts are turned back on for you.

I have double and triple checked this in various applications, turning Type 1 fonts on and off while I had a document open. Activate your Type 1 Helvetica fonts and the Apple fonts disappear from the available font menus. Deactivate the Type 1 versions and the Apple fonts are once again back. Same with the Helvetica Neue fonts.

If you'd rather be **completely sure** rather than relying on Suitcase's system to work, you may still want to remove Apple's fonts from the system folder and manually activate them from a different location when you don't need another version of Helvetica active.

General information for third party programs.

Microsoft Office 2016 used to have various issue if Helvetica Neue was absent, or was simply not the OS installed version. Version 15.39 and later don't require the presence of Helvetica Neue at all.

The Adobe apps aren't fussy. You can completely disable Apple's version of Helvetica and enable a Type 1 PostScript version. So that at least is nice for prepress and other users of Type 1 PostScript fonts. You can have the T1 PS fonts open for a project and still have all of your Adobe apps work.

Font folder locations - Using Font Book

macOS uses a folder priority to locate and open fonts. This list is in order from highest priority to lowest.

`~/Library/Fonts/`
`/Library/Fonts/`
`/Network/Library/Fonts/`
`/System/Library/Fonts/`
`/System Folder/Fonts/` (the OS 9 Fonts folder)

(Source: Apple Knowledge Base article [HT2435](#)).

Fonts activated from Suitcase or other third party font manager not located on the Mac in any of the above folders are given the least priority overall. You can see then that a version of Helvetica activated by your font manager will be superseded by any version of Helvetica located in any of the previously listed folders. In order to use your PostScript fonts activated by your font manager, all like named versions must be removed from the higher priority folders. (Suitcase Fusion can bypass this limitation and override the system fonts).

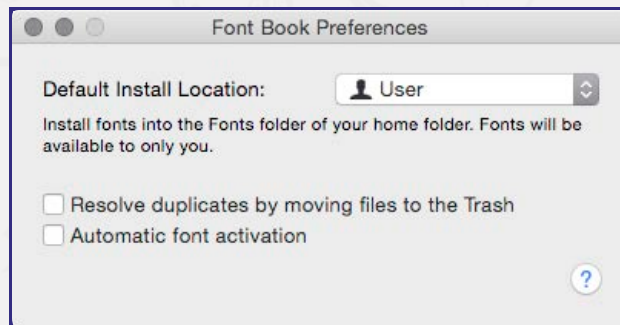
macOS's Font Book has preferential treatment here. By that, I mean fonts you activate using Font Book will get a higher priority over other

font managers as it uses the folders listed above to activate and deactivate fonts. This, if you're using the default method of adding fonts in Font Book. They are copied to the Fonts folder of your user account. As you can see in the list above, that folder has the highest priority.

The advantage of Font Book (besides being free) is that by knowing these rules, you can *try* to quickly force preference of one font over another of the same name by placing the font in a Fonts folder that has a higher priority, **although you should always avoid knowingly doing this**. A font conflict will almost always prevent either font from showing up in any application if both are active.

Font Book in El Capitan, 10.11

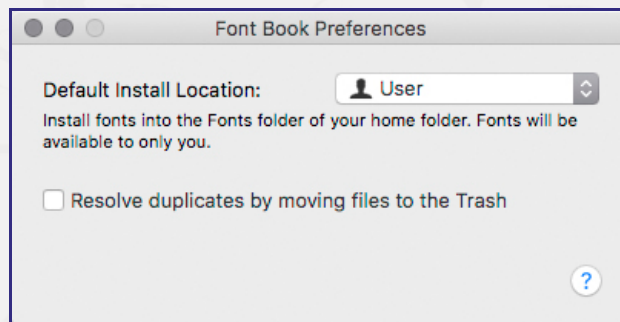
The preferences have been simplified. There is no longer an alert for system font changes.



Font Book preferences in Mavericks through El Capitan

Font Book in Sierra, 10.12 through Mojave, 10.14

And now even *Automatic font activation* has been removed from the preferences. Testing with both a Collection and a Library set confirms that auto activation has been removed from Font Book. Any fonts I disabled that were used in a test document were not turned back on when I opened the file again.



Font Book preferences in Sierra through Mojave

Resetting Font Book's database

At times, Font Book's database can become corrupt. Usually from opening too many fonts, or fonts that are damaged. When you activate fonts with Font Book using the original method, not only are all fonts you've ever activated copied to the Fonts folder chosen in its preferences, but those fonts are also added as entries in its database (Library sets do not copy the fonts, but still do become entries in the database). Font Book uses this database to keep track of which fonts are active and which are not. The more there are, the longer it takes for your Mac to start up as Font Book must compare each font to the information in the database during startup to determine if a given font should be on or off. If the database becomes damaged, it can take a very long time for your Mac to start up to the desktop. If it's really mangled, your Mac may not finish booting at all. Another symptom of a damaged database is not being able to activate or deactivate fonts from Font Book's interface.

Manually removing fonts from your system will also 'damage' Font Book's database, so to speak. Font Book does not correct itself when you manually remove fonts that are listed in its database; not even after a restart. It then contains links to non-existent fonts which causes the program to behave in the same manner as a corrupt database. You'll find that you cannot activate or deactivate certain fonts, or any at all.

El Capitan through Mojave each have their own methods for resetting Font Book.

To reset Font Book completely in El Capitan, 10.11.x through Mojave, 10.14.x, do one of the following.

1) Restart your Mac and immediately hold down the Shift key when you hear the startup chime to boot into Safe Mode. Keep holding the

Shift key until macOS asks you to log in (you will get this screen on a Safe Mode boot even if your Mac is set to automatically log in). Let the Mac finish booting to the desktop and then restart normally. This will clear Font Book's database and the cache files for the user account you logged into in Safe Mode.

2) Quit Font Book. Open the Preferences folder in your user account, put the following two files in the trash and restart:

`com.apple.FontBook.plist`
`com.apple.FontRegistry.user.plist`

The file `com.apple.FontBook.plist` keeps track of Font Book's general preferences and activated fonts, whether as a standard or library collection. `com.apple.FontRegistry.user.plist` keeps track of deactivated fonts. This method of resetting Font Book is faster than the first, where you have to restart twice.

Removing Font Book

As noted in section one, you really, *really* don't ever want more than one font manager on your hard drive at a time. So, if you don't use Font Book, it should be removed. You also need to remove its database. Leaving Font Book's database on the drive can interfere with other font managers, even if the Font Book application itself isn't running, or no longer exists on the hard drive. macOS will read this database (if it exists) during startup, or when logging into another account. Based on what that database says, macOS will fight to keep fonts active that the database says are supposed to be active, thus possibly preventing your preferred font manager from operating correctly.

Apple has protected both the Applications and Utilities folders for any program installed by the OS. This makes it a bit more difficult to remove Font Book, but not impossible. If you try to trash Font Book from the desktop, the OS states *Font Book can't be modified or deleted because it's required by macOS*. Don't take this message to mean Font Book is now an immutable part of the OS. El Capitan through Mojave say this about anything it installed when you try to them delete from the Applications or Utilities folders. It's hard to believe that Chess, Photo Booth, Stickies, or quite a few others are "required" by the OS. It will run just fine without them.

1) Follow steps 1 or 2 above under **Resetting Font Book's database**. The goal here is to remove Font Book's database.

2) Clear all font cache files from the system and restart. The Terminal method for this is explained at the very end of section 17.

Do not launch Font Book after this, or you'll just have to repeat these two steps.

In order to remove Font Book, you will have to either temporarily [disable System Integrity Protection](#), or start up to another drive. The latter is much easier since anything on a drive or partition that is not the startup drive can be removed without disabling SIP. You just need your admin credentials.

If you disable SIP and restart to the same drive you want to remove Font Book from, use either of these steps:

1) Open the Terminal application in the Utilities folder. Highlight and copy the following blue colored line, then paste it into Terminal. You can, of course, type the command in yourself; but be very careful to enter it *exactly* as shown. Unix shows no mercy for any misplaced or mistyped characters.

```
sudo rm -r /Applications/Font\ Book.app
```

Press Enter. You will be asked for your admin password. Type it in (Terminal does not return on screen what you're typing for a password) and press Enter. Font Book will disappear from the Applications folder. Close Terminal.

If you are using High Sierra 10.13.4 (and presumably from here forward), using the above Terminal command is now required. Even with SIP disabled, you cannot delete an OS installed app using the Finder. It's either this, or boot to another drive or partition so you can remove whatever you want from the non-startup drive.

2) If you prefer not to use Terminal, you can remove Font Book via the desktop. Highlight Font Book and press Command+I, or choose Get Info from the the top menu bar. Click on the lock at the lower right of the Get Info dialog box and enter your Administrator password. Expand the **Sharing & Permissions** heading if necessary. There are three permissions states, the first of which is already **Read & Write**. Change the other two, which are currently **Read only**, to **Read & Write**. Close the Get Info box. You can now move Font Book to the trash with no interference from the OS.

Re-enable System Integrity Protection after method 1 or 2.

Various other fonts you can move or remove

You can remove all of the OpenType fonts installed by InDesign or the Adobe suites so they aren't active when using those programs. InDesign or the Adobe suite opens them from its own font subfolder, so while they're available in the Adobe applications, other programs can't see them. The location for these OpenType fonts is `/Applications/Adobe InDesign CS/Fonts/`. Although more commonly, the location is `/Library/Application Support/Adobe/Fonts/`. You'll need to check both locations depending on which versions you've installed. Whether you install the Creative Suite, or the individual applications (Photoshop, Illustrator or InDesign; CS and CS2 versions), quite a few OpenType fonts will be installed in the latter folder. While no other applications or the OS will see those fonts, all of the CS and CS2 programs will. Move those fonts to another location for future use. Again, if you want to use those OpenType fonts for any program, you can activate them with Font Book, Suitcase, FontAgent or other font manager. You will also find a subfolder at `/Library/Application Support/Adobe/Fonts/Reqr/`. Do not remove that folder or any of its contents. The Adobe applications require those fonts and will not launch without them.

The Adobe CS3 through CC applications now install all OpenType fonts they come with into the `/Library/Fonts/` folder. These can all be removed. Those fonts required by the CS3 through CC applications are buried where only the Adobe programs can find them. There is no need for you to locate them. As in earlier versions, only the Adobe applications will see and use their required fonts.

Check the fonts in your `~/Library/Fonts/` folder. Again, look for duplicate fonts that conflict with the same name as those you prefer to use as a PostScript version and remove them.

If you have installed Acrobat or the free reader (older versions), you will find more PostScript versions of Helvetica and Courier in these applications' Support folders. Since none of them are in a location that will be automatically activated by macOS, they can (and need to be) left alone. Only those applications that installed them will use those fonts if they are not already active otherwise.

El Capitan through Mojave include fonts intended for iWork. Apple essentially gives you these fonts, even if you don't own iWork. In the `/Library/Application Support/Apple/Fonts/iWork/` folder, you will find various fonts. Because of the folder they are in, they are not automatically seen or used by the system. However, you can open them with any font manager in the usual manner; giving you some unexpected extra free fonts.

A few others you can copy include fonts within the Game Center app. Right click on Game Center and choose **Show Package Contents**. Drill down to **Contents > Resources > Fonts**. *Make sure to copy the fonts you find there* so you don't break the Game Center app.

There's a small treasure trove of fonts in Toast Titanium. Version 11 has 60 .ttf fonts. Right click on the Toast Titanium app. Drill down to the **Contents > Resources > Fonts** folder. Again, be sure to *copy* the fonts from this folder.

Did you ever find a folder named `/Library/Application Support/Adobe/SavedFonts/`, loaded up with a bunch of fonts identical to those in the `/Library/Fonts/` folder? Those are old versions put there by Adobe. When you install most of Adobe's older CS suites (and now with periodic major upgrades to the CC apps), the installers put a lot of OpenType fonts in the `/Library/Fonts/` folder.

Where the fonts in `/Library/Application Support/Adobe/SavedFonts/` come from is, for example, you upgrade CS5 to CS6. The installer looks for older versions of the same bunch of OpenType fonts it's going to install and moves those older versions into the SavedFonts folder.

I suppose rather than toss them, Adobe decided there may be just enough kerning or other differences in them that a shop may need to go back to an older version to prevent type reflow. If you don't think you need those older fonts, you can give 'em the boot.

Do you need a font manager?

Do you need to use a font manager at all? Actually, no. You can activate fonts by placing them into any of the Fonts folders of your hard drive mentioned in section 7 and removing them when you want those fonts closed. For convenience sake, I would suggest always using the common `/Library/Fonts/` folder. Fonts placed in this folder will be active to all users of that Mac. If there is some font you don't want other users of that Mac to have access to, place them in your user account fonts folder, which is located at `~/Library/Fonts/`. The most convenient way to use this method is to create an alias of the Fonts folder you want to use on the desktop. That way, you don't have to keep opening the hard drive and clicking down through the folders of the disk hierarchy to get to it.

There is a disadvantage to this method though. You run the risk of damaging fonts by constantly moving/copying/deleting them from the folder you're using to open and close them with. For these reasons, I do suggest using a font manager.

What is a font suitcase and why do we have them?

In decades past, computers had very limited amounts of RAM. This was partly due to the fact that RAM was extremely expensive. So most computers were outfitted with far less RAM than could be installed. This required the OS and applications to be as frugal with RAM usage as possible. Each font you activate takes up a small amount of RAM, so the Mac OS limited the number of fonts you could activate to 128.

Since OS 9, computers generally have had more available RAM, so this limit was increased to 512. Designers, however, often wanted or needed to have more fonts open at once than the OS would allow; especially for projects like a catalog. The solution (more like a workaround) was to enclose fonts in a suitcase. The suitcase itself was counted as only one item by the OS in the Fonts folder; so you could open dozens or hundreds of fonts by enclosing them in a suitcase. Having fonts enclosed in a suitcase was also the only way the Mac OS would recognize fonts before OS X.

Using suitcases has become unnecessary on today's computers with gigabytes of RAM and more capable operating systems. You see that with OpenType fonts. Each typeface of a font (italic, bold, etc.) is a file unto its own (those with a .ttf extension) rather than being placed together in a suitcase. Not that the usage of suitcases has ended. It's a very convenient way of keeping a font set together. Apple's .dfonts are suitcase files which contain the individual TrueType fonts for that family of fonts. There are four types of suitcase fonts currently in use:

1) Mac legacy TrueType suitcase fonts from OS 9 and earlier. These are 8 bit fonts limited to 256 characters, or glyphs. The suitcase can contain up to 999 individual TrueType or bitmap fonts. They can be of any font family. All data is stored in the resource fork of the fonts. It's not at all unusual to see a mix of fonts found in programs such as greeting card and banner makers. They'll give you a font suitcase named something like "Card Designer Fonts". You see just the one item on your desktop, but contains as many individual fonts as they put in it.

2) Type 1 PostScript fonts. These are two part fonts. One file is a suitcase containing all of the low resolution bitmap screen fonts. The rest are the outline printer fonts. As an example, here's Adobe Garamond:

Adobe Garamond

AGarBol
AGarBollta
AGarIta
AGarReg
AGarSem
AGarSemIta

The first file (which I highlighted in green) is the font suitcase of bitmap screen fonts. The rest are the individual outline printer fonts. Both must be in the same folder in order to work. When placed in a Fonts folder or activated with a font manager, the OS or font manager only looks in the suitcase for the available type faces.

If you have the printer outline font for the italic version of a font, but the screen font for the italic face is missing from the suitcase, then the italic font will not work. If you have the outline fonts without the matching suitcase, then none of them will work. In reverse, if you have the suitcase screen font for bold, but not the bold outline printer font; the bold font will show up as available in your font lists, but the printed output will be very low quality because the system will be forced to print the font from the low resolution bitmap font in the suitcase. Screen fonts in the suitcase that are missing the matching outline printer font are known as orphaned fonts. All data for Type 1 PostScript fonts is stored in the resource fork. They also are 8 bit fonts limited to 256 glyphs.

Beginning back in El Capitan, neither the OS or any font manager will allow you to activate only a Type 1 PostScript suitcase as you could before. This is good since all you have are the screen fonts and it shouldn't be considered usable. You must have both the suitcase and that font's matching printer outlines to use it at all. You either have a complete font, or you don't.

3) Apple TrueType .dfonts. They are essentially the same as the legacy Mac TrueType fonts from OS 9 and earlier with three major differences: (a) the data is stored in the data fork of the font rather than the resource fork (b) they are Unicode fonts and (c) like OpenType fonts they are 16 bit fonts which can contain 65,536 glyphs. However, they are not OpenType fonts as they have a different font table.

4) TrueType Collection. These fonts have a .ttc extension. Similar to a .dfont, it is a suitcase containing multiple TrueType fonts. Apple (for the most part) has moved to this suitcase type in Snow Leopard and later.

FFIL and LWFN. What are they and which is which?

There are a fair number of explanations to be found on the Web as to what FFIL and LWFN stand for, but in a nutshell they are the Type codes of the respective files. At the time Type 1 PostScript fonts were created, the Mac OS kept track of what type of file it was, and which application owned it, by its resource fork Type and Creator codes. The Type code of a font suitcase is FFIL. For the outline printer font, it's LWFN. macOS and OS X can still use them, but Type and Creator codes are being replaced by a more modern method known as Uniform Type Identifiers.

Many Type and Creator codes have a meaning attached to them while others don't; or at least not an obvious meaning. For Adobe Photoshop, an EPS file saved by the application has the meaningful Type code of EPSF; but the Creator code for any file saved from Photoshop is 8BIM. I'd never heard what that was supposed to stand for, if anything. So while there may actually be a phrase that FFIL and LWFN are an acronym for, from what I've been able to find they are nothing more than the assigned Type codes.

While you could call a .dfont an FFIL file since it is a suitcase, it's not a Type 1 PostScript suitcase. Having a different Type and/or Creator code is what helps the Mac OS keep similar items from being opened in the wrong application or defined incorrectly. In reality though, .dfonts don't normally even have Type and Creator codes.

A reader informed me that he was a beta tester for Photoshop v 1.01b. That version added 8 bit color support, which was huge at the time. The Creator code for Photoshop then became 8BIM, short for 8 Bit Image Map. A former employee of Adobe has confirmed the most commonly seen meanings for FFIL and LWFN found on the Web. They are: LWFN = LaserWriter Font, FFIL = Font File. Thank you both for your contributions.

External vs. internal font names

How many free TrueType fonts have you downloaded and found that many of them show up in your font lists with the name "New"? There is a reason for that. With all fonts the name of the file you see on your desktop has **absolutely nothing to do with the names that show up in your applications**. That is controlled solely by the font's internal name. Those free fonts were likely created by someone who (1) didn't know they needed to, (2) forgot to or (3) simply didn't care about assigning a proper internal name to their creation. Windows TrueType fonts in particular, even those included with commercial applications, are a constant source of this type of confusion. It's not at all unusual for a Windows TrueType font to have a file name of TT145B3.TTF, but shows up in your application as its internal name of "Bumblebee". To be fair, this can be a problem with any Mac font suitcase, also, because there is no easy way to tell how many fonts are in the suitcase or what names they'll produce in your font lists until activated.

Thankfully the designers of Mac fonts have almost always been kind enough to give the fonts they create descriptive file names, like the Adobe Garamond example above. Still the name of the suitcase is no true indicator of the font names that will show up in your programs, but historically this has been the case on the Mac.

Assigning the internal name is something the font's creator must do when using a font creation program such as FontLab or Fontographer. You can also use these programs to change the internal name of a font, or assign one where the original creator of the font did not. Like those three hundred free fonts you have that all show up in your font lists as "New".

What is a font conflict?

In order to understand how font conflicts occur, it was first necessary in the previous section to explain internal font names. The shortest explanation of a font conflict is that two or more fonts you have activated are declaring the same internal font name to the OS.

Fonts are actually little programs, or as once described by an Adobe representative, "Plug-ins for the OS". Fonts don't just sit on the hard drive waiting to be called on. When you activate them, each individual font takes up a small amount of RAM; which (among a few other things) is needed to load the internal name the OS displays in your applications. If you are opening a suitcase, more RAM is used because multiple names must be created at once; especially if it's a suitcase containing more than one font, such as a TrueType Collection, .dfont or Type 1 PostScript font can have. If the suitcase contains 30 fonts, it will open 30 separate tags in RAM.

So let's say you activate a font with a file name of Courier and its internal name is also Courier. Then you activate a second font with a file name of "Courier Plus", but the designer made its internal name Courier. You now have two fonts declaring the same internal name of "Courier" in RAM with the obvious conflict; more than one active font saying it is Courier. When you go to choose them in your application, how can the OS or the application possibly know which one you mean to use? The answer of course is, they can't.

Various things happen when you have font conflicts. Sometimes the font you just opened with the same internal name will take precedence over the one that was already open. Other times the font that was already active with that name will be the one to continue to show up in your programs, the new one won't. Rarely, if ever, will you see more than one font with the same name show up in your lists. More likely, active fonts with duplicate internal names won't show up in any program. Quark XPress is very good at having fonts disappear from its lists when there is a conflict. This is not a bad thing and I wish more applications would do that. It's essentially letting you know immediately you have a font conflict by not showing you a font you're expecting to see in its list of available fonts.

Conflicting internal names is exactly what the font problem is between Apple-supplied versions of Helvetica in El Capitan through Mojave and older Type 1 PostScript fonts. Apple gave almost every individual typeface exactly the same internal name as the Type 1 PostScript versions. Since the OS protects the system fonts from being disabled, you can't open your preferred versions of Helvetica. This can be circumvented, however, by following the instructions in Section 6.

Are you a good font, or a bad font?

Other than not being able to see a font you expected to, what is actually happening behind the scenes when you have a bad font? There are

more problems that can occur with fonts than just conflicting internal names.

One issue is damaged fonts. Just like an application, fonts take up space in RAM. Typically it is a small amount of RAM; however a damaged font, just like a damaged program, will do things it's not supposed to. With fonts the most common issue from becoming damaged is a memory leak. A font may normally only take up a few Kbytes of RAM, but a damaged font can step outside the amount of RAM the OS thinks it is supposed to be using and take more. When this happens the font may overwrite data being used in adjacent RAM by another active application or the OS itself. Suddenly, data the OS or an application may need, or be looking for is gone; overwritten by the damaged font. Disabling the font *might* release the RAM back to the assigned program or the OS, but the damage is already done. In the case of an application, you will need to shut it down and relaunch it in order for it to behave correctly again. The OS will likely require a restart.

All other font issues are in how they are built. Free fonts are the number one source of these font problems. Free fonts known to be bad are **Alien League** and **Brady Bunch**. Both have an incorrect internal name. There are several name fields that need to be filled in by the font's creator. Among them are the PS Font Name, Full Name, Menu Name and FOND Name. In the case of both fonts mentioned here, three of the fields have the same name as what the font is supposed to be, but both have **Arial** in the Full Name. So both conflict with the real Arial font. Another 'bad font' is **Radioactive**. It has an incorrect PS Font Name which causes it to conflict with Times New Roman. These are three I know of, but I'm sure there are many more.

If you want to see the effects of this (it's harmless, but annoying), download the Brady Bunch font and activate it. You can easily find this font by using "font+Brady Bunch" (without the quotes) as a web search. Google makes heavy use of Arial in the text it returns for a search. When you display such a page, Brady Bunch gets in the way of the regular Arial (Roman) font. Arial Bold will still be correct since that's a different internal name and a separate font, but all regular Arial text will display in the Brady Bunch font. Now that you've seen it, deactivate the Brady Bunch font, toss it in the trash and everything will return to normal.

Newer info on the Alien League font: I don't know if it's due to this article, but Alien League has been updated and can now be downloaded from *most* sites as a correctly functioning font. There are still bad copies out there, but they are actually getting very difficult to find. You can't go by the name of the font since both a bad copy and a fixed version are named alien5.ttf. Without a font editor to view the internal names with, you can't really know which is which without activating it and seeing if Arial suddenly gets trumped by Alien League.

The Brady Bunch font has also been fixed. Lots of copies of the old bad version are still out there, but if you want a repaired version, download Brady Bunch Remastered.

With any font, there are multiple common errors they can possibly have. These issues can include outlines that make up the shape of a character having far more points than necessary to define a curve, having segments which overlap, paths running the wrong direction, and/or have stray anchor points. If you are unfamiliar with vector drawing, you draw a shape by placing down points. Between the points are [Bézier](#) line segments. You continue to draw your shape until you come around to your first point and close the path by connecting the last anchor point to the first. A path in Photoshop is an example of a vector outline, as is any such shape drawn in Illustrator. Fonts shapes are drawn the same way.

Free fonts may be poorly designed (and thus earn a 'bad font' rating) due to the lack of effort by their creator. Many are done by drawing characters by hand with pen and paper, scanning them into the computer and then using a tracing program to create the paths. These result in rough fonts using hundreds to thousands more anchor points than necessary to create the outline shapes. Problem is, most people make no attempt to clean them up, but instead just drop the resulting paths into the font they're creating. Yes, it saves a *lot* of time creating your font, but the end result is the junk they are.

Unnecessary points make the vector to raster processor work a lot more than it should have to when displaying the font to your screen and printing. Granted, computers do this so fast you'd have a hard time measuring the time difference between a well drawn font and a poor one, but it's still an indication of quality over a slap-it-together font. Overlapping segments (Bézier lines) cause a break in an outline that shouldn't exist. If a path is running the wrong direction (the outer path must be created in a clockwise direction, and the next inner path counter-clockwise), various things happen, depending on the character's shape. For example, take the letter 'A'. Depending on correct or incorrect drawing of the outer and inner paths (for the hole in the middle), the character may display as just a solid triangle where the hole is, but invisible where it should be solid. A stray point is an anchor point that connects to nothing. Stray points can be particularly bad for RIPs. When the RIP comes across one, it sees the point, but it has no other instructions then what to do with it. It can't draw anything since a single anchor point goes nowhere and isn't a shape unto itself. It's not a line, so the RIP can't draw one. The RIP can't fill a shape since there is none. The automated error handling of a RIP may be able to handle the problem, but otherwise may cause it to crash.

While there are a handful of good free fonts out there, the vast majority are worth exactly what you paid for them. For an excellent source of free fonts, head to [Font Squirrel](#). I've downloaded and examined quite a few randomly chosen fonts. All are very well made. The owners of Font Squirrel vet the fonts you find there, so only the highest quality free fonts are made available.

Fonts you can and cannot use in macOS

Fonts which can be used in macOS:

Apple Data Fork Fonts (.dfont) - A Unicode TrueType suitcase variation which allows for 65,536 characters per font.

Color Fonts (.otf .ttf and .ttc) - Until recently, you haven't had this much fun with fonts! Think of them as Type 3 PostScript fonts on steroids. While those could only use simple fill patterns, shading and solid colors, color fonts can be darn near anything. You've actually been using one for a while and maybe didn't even realize it - Apple Color Emoji.

There are four types of color fonts. The chart shown on [this site](#) checks off which types an OS directly recognizes. As of Mojave, macOS and Safari (version 12) now support three of the four color font versions, including SVG (Scalable Vector Graphics) fonts. Apps that don't support SVG color fonts will see and use them, but only as a standard font.

Which apps and web browsers currently support color fonts are also listed on the site. For example, if you have the CC 2018 apps, Photoshop, InDesign and Illustrator can use both SVG and bitmap color fonts. Quark XPress 2018 fully supports all four types of color fonts.

Adobe has the Trajan color font active in Photoshop CC 2017 and later. You can add it to TypeKit for use in your other Adobe apps by heading to [this site](#).

So, where can you get color fonts? Lots of places! There's a link to download a .zip file of 105 free color fonts [here](#). Another 70 or so [here](#) (the image at the right of each one shows how it will look in color). Or, buy them [here](#). Lots to pick from and very reasonably priced.

Feeling creative? [Fontself](#) sells a \$49 plug-in that works with Photoshop and Illustrator CC 2018 to build your own.

For now, Safari does not support displaying color fonts within the browser. In order to view some of these sites correctly, you need to use Firefox or Chrome.

The question you may ask. Are these just a gimmick? For those in prepress, maybe not a whole lot of interest. But for designers, they're gold! Think of the time you had to spend in the past taking a plain font shape and then running it through lots of filters and other steps to get a chiseled look, metal, brush strokes, wood, etc. Now you just type!

I can see these fonts becoming commonplace and heavily used in a fairly short amount of time.

Mac legacy TrueType - Also a TrueType suitcase font, normally with no file extension. 8 bit fonts limited to 256 characters.

Mac Type 1 PostScript - Paired fonts comprised of a suitcase of screen fonts and the individual outline printer fonts. Most often without any file extensions. Sometimes the suitcase of screen fonts will have an extension that helps define that it is the screen font suitcase, but aren't otherwise necessary. The file name extensions found on some suitcase files vary between .bmap, .scr and a few others. Originally, 8 bit fonts with a 256 glyph limit. Encoding options were updated later to allow for some extension of the number of allowed glyphs.

An example of a Mac Type 1 PostScript font can be seen in section 10 with Adobe Garamond. As you can see, "paired" doesn't necessarily mean only two items. There is always just one suitcase of screen fonts, but there can be any number of printer fonts. Each printer font will have at least one matching screen font within the suitcase.

Multiple Master (Mac OS X 10.2 and later only) - A Type 1 PostScript variant. Rarely used in production. While older versions of Preview and the Acrobat Reader do depend on certain Multiple Master fonts for their operation, they are no longer produced by Adobe and have been declared obsolete.

OpenType (.otf .ttf and .ttc) - Unlike the majority of fonts between the Mac OS and Windows, OpenType fonts are Unicode and have the advantage of being cross platform compatible. They are 16 bit fonts capable of having up to 65,536 characters. Files with a .otf extension, which contain only CFF data will always be a PostScript font. TrueType OpenType fonts will have either a .ttf or .ttc extension. Unfortunately, this is not always true. The OpenType [guidelines](#) allows developers to use .otf for a TrueType structured OpenType font if they wish. So just by looking at it, it is impossible to know whether or not the .otf font you have is actually a PostScript font.

'Impossible' sometimes has a workaround connected to it, so there are ways to find out if the .otf font you have is actually a PostScript font. Activate your OpenType fonts and then launch Quark XPress. In the list of fonts, Quark separates OpenType fonts by structure. A TrueType font will have the standard green and black OpenType icon. PostScript OpenType fonts will be shown with a red and black icon. Suitcase Fusion also notes which type of fonts they are. A PostScript font will be shown in the Type column as **OpenType - PS**, and a TrueType version as **OpenType - TT**.

OpenType Variable (.otf .ttf and .ttc) - These are essentially OpenType versions of the old Multiple Master fonts. But, they're also a lot

easier to use, in apps that give you the necessary controls. Adobe has a good overview on them (and includes a nice animated GIF of one in action) on [this page](#). If you want to try a few, Photoshop CC 2018 lists six variable fonts you can use while in the app. You won't find them anywhere on your drive. They are live fonts, but only appear in Photoshop from wherever Adobe hides them.

Windows TrueType (.ttf) - The Windows version of the original 8 bit, 256 character TrueType fonts.

Windows TrueType Collection (.ttc) - Windows version similar to Mac TrueType suitcase fonts. They can contain more than one TrueType font in a single package. This is again referring to the original 8 bit version, even though the same .ttf and .ttc file extensions are used for the OpenType versions.

Fonts which cannot be used in macOS:

Apple bitmapped fonts from OS 7.5 and earlier

Linux Type 1 PostScript - Paired fonts with .pfa and .afm extensions.

Type 3 PostScript - A fairly short lived PostScript variant. They allowed for the full use of the PostScript language such as shading, color and fill patterns. However, they lacked hinting, an important typography attribute. Obsolete.

Old, very old third party bitmapped fonts. Way back before TrueType or Type 1 PostScript fonts. There were no outline vector fonts for printing. Each point size for a typeface had to be built as a high resolution bitmap file from a companion program, or purchased separately. Good riddance.

Web Fonts - OpenType and TrueType fonts can be served from a remote computer to function as web fonts, but there are three other styles that only work in your web browser. Those end in the .eot, .woff, and .svg file extensions. macOS, OS X and any font manager will pay no attention to these font types if you try to use them directly on your Mac.

Fonts which can "sort of" be used in macOS:

Windows Type 1 PostScript - Paired fonts with .pfb and .pfm extensions. Normally, these fonts have never been able to be used on the Mac; not in OS9, OS X or macOS. Technically, you still can't. You can't open them with any font manager or use them by directly placing the fonts in one of the standard Fonts folders. But the Adobe applications (Photoshop, Illustrator, InDesign) starting with the CS2 suite can indeed read these fonts. Place them in the [/Library/Application Support/Adobe/Fonts/](#) folder and you will be able to use Windows Type 1 PostScript fonts with (and *only* with) the Adobe programs. The CS4 through CC versions do not create the Fonts folder mentioned, but all you have to do is go to the [/Library/Application Support/Adobe/](#) folder and create a new folder named **Fonts**.

In InDesign CC 2015 and later, there is now an empty Fonts folder in the [/Applications/Adobe InDesign CC 2015/](#) folder for just this purpose. Use this folder if InDesign CC 2015 is the only app you need to see Windows Type 1 PostScript fonts. For no reason other than it's faster to get to.

An example of a Windows Type 1 PostScript font is as follows:

DSSCR____.PFM
DSSCR____.PFB

This particular font is Dorchester Script MT. All Windows Type 1 fonts consist of two files for one complete typeface. A .pfm file for the font metrics, and a .pfb file for the binary data. You must have both in order for the font to work. If there were a bold version of this font, you would have another uniquely named matching pair of .pfm and .pfb files.

Font manager reviews

I've used (or at least tested) every font manager mentioned in this article. To help you make a more informed choice as to which one you may want to use, I'm going to list the pros and cons of each one as I see them. These are not exhaustive reviews, but focus on the more common features or omissions that make a given font manager easier or harder to use. Reviewed here are the most current versions available at the time of writing, so some features may not be available depending on the version you are using. They are; FontAgent 7.1.2, Font Book in macOS 10.14.x Mojave, FontExplorer X Pro 6, Suitcase Fusion 9, RightFont 3.2.3, and TypeDNA 2.7.1.

If there is a font manager not mentioned here, it has either been discontinued (in which case I will remove the review), or is too simplistic to bother reviewing. A fair number of commercial font packages will include such basic font managers. They turn fonts on and off, and that's pretty much it.

FontAgent

Pros:

- 1) Copies all fonts you activate to a separate location in your user account. Therefore, even if you remove the original font or no longer have access to it, you can still activate the font from its working folder.
- 2) Can create full sets that include font names already in other sets.
- 3) Reliable automatic font activation.
- 4) Font Savant technology virtually ensures auto activation will open the correct version of a font. MagicMatch will give you close optional choices when the original isn't available.
- 5) Now includes Smasher.

Cons:

- 1) Fonts cannot be activated in place. Everything is copied to its working folder.
- 2) Does not inform you of a conflicting font that is already active in another set unless you're using auto-activation.

Font Book

Pros:

- 1) It's free.
- 2) Can create font sets.
- 3) Resolves font conflicts. Much better control in Lion through High Sierra than previous versions.
- 4) Can deactivate fonts in the `/Library/Fonts/` and `~/Library/Fonts/` folders to prevent any application from seeing them.
- 5) Can create Library sets to open fonts in place.
- 6) Has an automatic activation feature.

Cons:

- 1) Different library sets can contain the same fonts, but they must be linked from exactly the same location in order to activate them.
- 2) Resolving fonts always favors those in the `/System/Library/Fonts/` folder. This makes it impossible to activate another type of font with the same name without having to first manually remove the conflicting font in the System folder.
- 3) The conflict resolution function, whether using automatic or manual, can and will remove fonts from the main Library folder. Handling of this issue is much improved in Yosemite and later.

FontExplorer X Pro

Pros:

- 1) Can create full sets that include font names already listed in other sets.
- 2) Can deactivate fonts in the `/Library/Fonts/` and `~/Library/Fonts/` folders to prevent any application from seeing them.
- 3) Can activate fonts in place. No need to create copies of the fonts you activate.
- 4) Can also be set up to behave like FontAgent and copy all activated fonts to a working folder to be managed from there.
- 5) Has an automatic font activation feature.
- 6) Let's you decide which font to activate when opening a conflicting font.
- 7) Very good at catching faulty fonts.

Cons:

- 1) Auto activation only looks for the correct internal font names. You could activate the wrong version of a font on an existing project.

RightFont

Pros:

- 1) Simple, straightforward interface. Even simpler than Font Book, which may appeal to novices.

Cons:

- 1) No font sets.
- 2) No method to add duplicate fonts to handle modified versions.
- 3) Very long, monolithic display of fonts to search through.

Suitcase Fusion

Pros:

- 1) Can create full sets that include font names already in other sets.
- 2) Can activate fonts in place. No need to create copies of the fonts you activate.
- 3) Has a reliable automatic font activation feature.
- 4) Font Sense technology virtually ensures auto activation will open the correct version of a font.
- 5) Let's you decide which font to activate or keep active when opening a conflicting font.
- 6) Individual fonts in a suitcase can be activated or deactivated rather than all or nothing.
- 7) Can deactivate fonts in the `/Library/Fonts/` and `~/Library/Fonts/` folders to prevent any application from seeing them.
- 8) Has a vault option if you prefer to store copies of fonts that have been added.
- 9) Can control conflicting system fonts automatically.
- 10) Quickmatch feature helps you locate fonts which are similar to each other.

Cons:

- 1) Conflicting fonts cannot be acted upon individually.
- 2) Doesn't catch as many bad fonts as FontExplorer X Pro.

TypeDNA

Pros:

- 1) Can create Groups (sets) that include font names already in other sets.
- 2) Activates all fonts in place. No need to create copies of the fonts you activate.
- 3) Let's you decide which font to activate or keep active when activating a conflicting font.
- 4) Can deactivate fonts in the `/Library/Fonts/` and `~/Library/Fonts/` folders to prevent any application from seeing them.

Cons:

- 1) Groups are difficult to set up.

Font Book is included with macOS, so you can play around with it as much as you want. FontExplorer X Pro, Suitcase Fusion, FontAgent and TypeDNA all have versions you can download and use as fully functional software for 30 days. Plenty of time to run them through the paces.

Suitcase Fusion has solved the issue with the Helvetica system fonts. You don't need to move, or remove any fonts from the `/System/Library/Fonts/` folder if you don't want to. When you activate (for example) a Type 1 PostScript version of Helvetica (which will conflict with Apple's versions) Suitcase Fusion automatically turn the system Helvetica fonts off! When you deactivate your preferred version of Helvetica, Suitcase automatically turns the Apple versions back on. See section 6 for more details.

Part of the original point of this article was to work around the inability of most font managers to disable fonts in the two main Fonts folders from their interface. By manually reducing the fonts on your system to only those listed in Section one, it won't matter that you can't control the remaining fonts; and you shouldn't be trying to deactivate them anyway. That makes it a plus for Suitcase Fusion, FontExplorer X Pro, Font Book and TypeDNA to be able to deactivate fonts in the `/Library/Fonts/` or `~/Library/Fonts/` folders from their respective interfaces; especially if you are uncomfortable about moving fonts out of those folders.

I consider full font sets a very big plus. When you have multiple projects going at once, you want to have a single set for each project that includes every font it uses, not just those that don't already exist in another set. That makes your stop to your font manager a quick and painless process when you can simply turn off set three and turn on set ten. No need to search the other sets for fonts you still need activated. I also consider activating fonts in place a major plus. If you can activate a font right from where it is, why bother copying it to another folder as Font Book (standard set) and FontAgent insist on doing? The only advantage to that is if the fonts reside on removable media when you first activate them, so you may not have access to the originals later. But to avoid that, all you have to do is copy the fonts to your hard drive first. It's not like they take up a lot of space.

Font Book in El Capitan through Mojave now allows you to open fonts in place by using Library sets. But there are still limitations to how you can add fonts. See below.

So, here are my recommendations based on what I consider important.

Please note that some of these descriptions may not apply to the version of a given font manager you are using. Other than

some older notes I leave for comparison, they are based on the versions which were current at the time of writing and may be a little, or drastically different from older versions.

1) FontExplorer X Pro: I didn't think I'd ever need to do this. After a very long devotion to Suitcase, I've switched font managers.

FontExplorer X Pro supports full font sets. Full choice of which font to activate or keep active when you try to activate a font that conflicts with one that's already on. Allows you to deactivate fonts in the `/Library/Fonts/` or `~/Library/Fonts/` folders. Activation of fonts in place.

During its initial setup you get a check box to "Organize Font Files". This is where you determine how you want FontExplorer X Pro to handle your fonts. If you leave the box unchecked, then fonts are always activated in place. Turn the check box on and it operates in the same manner as FontAgent, by copying or moving the fonts you add to its own storage space in your user account. If you change your mind later how you want FontExplorer X Pro to handle fonts, you can open its preferences and make your changes.

Unlike FontAgent, you can keep fonts that have been removed from the interface in storage, if you're using that option. A check box in the Advanced preference settings allows you to determine if items removed from the interface also removes the associated fonts.

Under System Fonts of version 5.x and 6.x, the root Library folder is referred to as All Users.

The default set, *Language Supports English* is very confusing. As an example, I duplicated a handful of fonts from the `/Library/Fonts/` folder and added them as a new set. As expected when using the option to *Import font as duplicate*, and then applying the same choice to all remaining fonts in the set, FEX added all of the duplicates as separate entries. As you would also expect, when you turn that set on, you are told all of these fonts are already active in the `/Library/Fonts/` folder. I then chose to have FEX deactivate the currently activated fonts and applied that choice to all other fonts in my duplicate set. The duplicates in the `/Library/Fonts/` folder were then deactivated.

All good so far. Just what you would expect the font manager to do. The problem is, after you've turned any of these duplicate sets on or off, the *Language Supports English* set continues to show those fonts as being off, even if they're enabled in any set they're listed in. Unless you have some great need for this set to exist; like you routinely work with multi language documents and it actually is at least somewhat important for you to know if a given font supports English encoding, I would suggest right clicking on that set and removing it. For no reason other than to avoid the confusion of having it give you the impression fonts are disabled or enabled. Either that, or just don't click on it. It's not there to control fonts. It's just to let you know which fonts on your system support English encoding. Dark gray check box - yes. Light gray - no.

FontExplorer X Pro has a much improved interface (though still very busy compared to Suitcase) and new features. Compared to the original free versions, I find it less cluttered than before, and therefore easier to navigate.

Auto activation is reported to be spotty. Fonts that should open with a document don't. If it still relies only on seeing the correct font name, then you could open the wrong version of a font and not know it. Suitcase Fusion and FontAgent have a method of avoiding this possibly very expensive error.

This is never an issue for me as I simply don't trust any auto activation feature, no matter who writes it. There's too much money at stake in large print shops if the wrong font is used on a project. I make a set for each job I work on and put the fonts for that job in its own set. When I'm working on a project, only that set is on.

FontExplorer X Pro has a large number of options to fully control your fonts. Overall, it is currently the best font manager available.

2) FontAgent: Auto activation is very reliable. Stable program that rarely crashes. Copies all activated fonts to another location, which I don't find necessary. FontAgent 7 now uses Sets, similar to Suitcase Fusion or FontExplorer X Pro. Much less confusing to have it behave the same as its competitors. The makers of FontAgent have put a lot of work into version 7. It's faster, has a much nicer layout, and is easier to navigate, understand and use.

That's the good news. For professionals, it's still a dangerous font manager to use. The one main issue I've mentioned for previous versions is still its greatest downfall. And that's being able to activate multiple, conflicting fonts without the user ever once being informed by the app. And even that isn't consistent. There's only a preference to inform you of conflicts through auto-activation. Any sets you manually turn on that produce a conflict tells you nothing. As in earlier versions, if you have duplicate fonts in multiple sets, they all shown green (activated) when you turn any one of them on. I've long wondered, is FontAgent simply being smart enough to recognize the fonts are identical? So I tested it to find out.

As with all previous versions, FontAgent creates a repository in your user account of every font you add, saving its own copies of all of them. This time, I took a font and intentionally stripped a bunch of glyphs out of it and saved a copy to the same name. All internal names were the same, and since the fonts were now physically different, it should produce a conflict error. At least, that's what you'd think.

If the fonts are truly identical, then no matter how many times you add the same duplicated font in different sets, FontAgent keeps one copy

of the font, and one only, in your user account. So yes, it does realize they are 100% identical, which is why every single instance of a given font turns green when you activate it in any entry. So far so good, since that makes sense, once you understand why it's doing that. Next, I added an unmodified font in one set, and then the modified version in another set. Turning one or the other on does not turn both green. Yes, again. FontAgent realizes they are not identical and has copies of both in the user account.

This all sounds logical and good. Right up until you turn one font on, and then the other. At this point, FontAgent simply should not let you do that. But you get no notice you've turned on not just a conflicting font, but a physically different one. Clicking on each one in the interface does show a preview for each "active" font in the bottom pane. And since I really hacked the modified one, it's clear glyphs are missing. So it should be impossible for both to be active.

In your apps, as you would expect, only one instance of that font name appears in the menus. Which one do you get? Since you now have a font conflict, it's essentially random. Sometimes the first font you activated is the one used, and other times it's the second one. So we have the same issue as previous versions. You could end up using the wrong version of a client supplied font, which could result in the loss of hundreds, or thousands of dollars at your company's expense.

If you remove any fonts or sets from the interface, FontAgent will also delete the related fonts from its storage location in your user account. You don't get a choice of whether or not the fonts are removed.

FontAgent's Font Savant technology, which is similar to Suitcase Fusion's Font Sense, makes FontAgent a strong contender. Having auto activation open exactly the same fonts as before is extremely important in a production environment, saving you from both lost time and income because the wrong version was used. When the original font cannot be found, MagicMatch will show you the closest alternative it can find.

Overall, FontAgent 7 is a very nice improvement to previous versions, but its biggest failure is that it may not warn you that a conflicting font you've turned on isn't the one previously used in a document. This can be a tremendously costly error in production.

3) Suitcase Fusion: Suitcase has been around for a very long time and it shows by its polished, easy to use interface. Excellent font activation that will not let you create font conflicts. Top notch auto activation with Font Sense to virtually guarantee the exact same font will be activated that was used before in a document. Full font sets. Choice of which font to activate or keep active when you try to activate a font that conflicts with one that's already on. Activation of fonts in place. Suitcase Fusion is incredibly stable. It is almost impossible to make it crash, even with the worst fonts.

What I wrote about version 8 earlier was true. Despite being a fresh rewrite, it was an incredible upgrade. But, something happened along the way. It used to be fast. Very fast. Now it's incredibly slow. It takes a long time to launch or close. It takes even longer to activate or deactivate fonts. And I mean *really* long. Even how I tried to activate fonts made a difference. Activated in place, Temporary or Permanent made no real difference. Most fonts would not appear in Office 2016 or other apps. Activated using the Vault, Temporary failed to activate most fonts. Permanent did much better, but still left quite a few unavailable. Basically, as you went down the alphabetized list in Office, it would (for instance) cut off at the T's and no other activated fonts below that point were accessible. This despite all of the fonts being marked as active.

It turned out (after much testing to find out what the "it" was), the basic issue is that Suitcase 8 is glacially slow. If you wait long enough after activating a large number of fonts, it eventually activates all fonts it had already marked as active. And then only if you use the Vault and Permanent activation. If you open MS Office too soon, it will only show what fonts have been activated so far. Since Office will not pay attention to any fonts that are activated after it's already running, you have to close Office and launch it again.

Suitcase 9 hasn't improved anything. In testing, it seems to actually be worse. I tried the same steps mentioned just above for Suitcase 8. Using 2,500 OpenType fonts for the test, I activated the fonts in place. It took over 5 minutes just to add the fonts while I sat and watched. Trying to activate the fonts either as Temporary or Permanent took so long to do, I wondered if the app was hung. No, it was still going, but extremely slowly. Less than a tenth of the fonts it said were active would appear in an app. As the Core continued to *slowly* activate fonts, they'd appear two or three at a time every 10-15 seconds.

With over 2,000 fonts to go, I finally force quit Suitcase 9, relaunched it, quit the Core and replaced the database with a new copy so I could start fresh. This time, I turned on saving fonts to the vault since that worked much better in version 8. Not this time. It was equally slow, such that I again thought it had simply stopped. Activation also wasn't any better than activating in place.

Now, I know I'm not working on a screamer of a Mac here with a 2010 Mac Pro, but I did the same test with FontExplorerX Pro. It added all of the fonts in place in about 30 seconds. It activated all of them in roughly 15 seconds. All of the fonts were available in every app. Disabling the entire set took less than 10 seconds. FontAgent worked just as well.

Granted, such a test is a bit extreme. Folks don't normally have anywhere near this many fonts open at a time. But the idea is to stress each font manager I test and see how it does. If you activate a couple of hundreds fonts or less at a time, Suitcase works as well as any other font manager. Give it too many though, and it practically dies.

What happened? Unknown to me. The same version 8 that was very fast is now very slow. Version 9 tested worse. Either Extensis goofed something up, or the continuing changes to macOS has caused a compatibility issue. I did have a nice discussion with one of their staff, exchanging a lot of information, testing and setup details. But I also can't expect them to find, fix, test and distribute a possible fix in a snap. It may be a while before we see an update.

Suitcase Fusion's handling of conflicting fonts is pretty simple. You choose in its preferences whether fonts you've just activated get turned on and the current active font off, or the existing font in use stays on. It's an all or nothing choice; so you can't do it on a font by font basis as you can in FontExplorer X Pro, or earlier versions of Suitcase. I always turn sets on and off, with each set having every font I need for a project in them; so this is never an issue for me. Others may find it constricting.

When Extensis acquired DiamondSoft, they were able to add the best features of Font Reserve to Suitcase, hence the name Fusion. Notably from Font Reserve, Suitcase inherited Font Sense and the ability to deactivate individual fonts within a font suitcase. Before, you had to either enable or disable the whole suitcase. That feature disappeared for a few versions of Fusion, but is back in version 8.

Suitcase Fusion can control fonts in the `/Library/Fonts/` and `~/Library/Fonts/` folders. I would recommend you remove all but the most basic fonts listed in section one to get your system down to only the fonts it needs in the `/System/Library/Fonts/` folder. It's up to you then whether or not you want to empty out the `/Library/Fonts/` and `~/Library/Fonts/` folders since they can be controlled from within Suitcase Fusion.

The interface and preferences are even more intuitive than the original OS X versions or the initial version of Fusion. Extensis has continued to produce a font manager that is easy and almost goof proof to use. While its feature set is still top notch, the current version is too slow and has other activation issues. With version 9, things are even slower.

4) Font Book: Simple, but effective.

Using the original set method, all activated fonts are copied to the Fonts folder of the active user account (default). You can change it to copy all added fonts to the `/Library/Fonts/` folder. Library sets do allow you to activate fonts in place without fonts being copied to the Fonts folder designated in Font Book's preferences. There are problems, however. While you can create multiple Library sets that include the same fonts, they *must* be from the exact same location. Adding identical fonts from different locations will add them to the Library set, but then you can't activate either one once they've been deactivated.

When you have conflicting fonts, those already installed in the `/System/Library/Fonts/` folder always win. The opposite fonts will always be the ones disabled. All others in the `/Library/Fonts/` or `~/Library/Fonts/` folders can be controlled, but you must do things in the correct order. If you tell Font Book to automatically resolve conflicts, the fonts which are currently active are always the ones to remain active. If you want the conflicting font enabled, you must first deactivate the active font. Then you you can activate the opposite font. This is true even in the conflict resolution interface.

Be very careful if you turn on Font Book's preference to move conflicting fonts to the trash. For example, let's say you have a font in the `/Library/Fonts/` folder deactivated, and an identical font active from somewhere else (which you can do in a Library set). Now turn the font in the `/Library/Fonts/` folder on. Since the font which was already active is always the one to remain on, when you resolve the conflict, the font in the `/Library/Fonts/` folder will be deleted! Not kidding. Right out of the main Library folder. So not just you, but *every* user account on that Mac loses access to that font.

Font Book's conflict resolution screen in El Capitan through Mojave actually improved, and does what you would expect! As an example, I duplicated Arial Regular from the `/Library/Fonts/` folder to the desktop, then activated it in a new standard font collection. This copied the font to my user account Fonts folder. Font Book of course declared there was a font conflict. Clicking the button to automatically resolve the conflict did what most users would want it to do. It deactivated the Arial font in the `/Library/Fonts/` folder, allowing the font you just asked Font Book to activate to be the active font.

While Font Book doesn't tell you which conflicting font it disabled, you can click on each set and then expand the font in question. Clicking the expansion arrow next to Arial under Computer showed that Regular was off, while Italic, Bold and Bold Italic were still active, proving that the duplicate I asked Font Book to activate was indeed the version in use. So, thank you Apple for this tremendous improvement in conflict resolution. The user can now easily determine which version of a conflicting font is on or off.

All of those extra Asian fonts that show up in Font Book's interface, which aren't anywhere in the various Fonts folders, are buried in the Font Book application package itself. They used to be called Stub fonts, and have returned in Sierra through Mojave as subset fonts (same thing) buried deeply on the drive. Leave them alone.

The All Fonts heading has finally been fixed. At least in El Capitan through Mojave. Collections and Library sets now both appear in that list along with system installed fonts. Before, fonts you added to Library sets would not appear under All Fonts. The only fonts that will not be shown now are those Font Book intentionally hides. Such as the system's San Francisco fonts.

A real killer for Font Book is how incredibly slowly it imports fonts. I've never been impressed with Font Book, and I think that comes

across in my review, but I wanted to see how much of a difference there was between Suitcase Fusion 8, FontExplorer X Pro and Font Book in High Sierra. I brought a total of 2,500 various font types into each app (3,000 individual typefaces). I used activation in place for each so no font copying was involved. This is the time it took in each just to import the fonts. Suitcase Fusion 8 - a bit under a minute. FontExplorer X Pro - less than 30 seconds. Font Book - one and a quarter *hours!* I have to say, I don't know why anyone would want to use Font Book, or why Apple hasn't junked the current version entirely and started over.

Font Book's main downfall is how easily its database is corrupted. When that happens, you suddenly can't activate or add some fonts, deactivate those which are already active, have fonts show up in some applications but not others, etc. The only fix is to restart in Safe Mode and back again with a normal restart to clear Font Book's database. In later versions of the Mac OS, you can open the Preferences folder in your user account, put the files [com.apple.FontBook.plist](#) and [com.apple.FontRegistry.user.plist](#) in the trash and restart. This is an okay font manager for light home use, but not a production environment.

5) TypeDNA: Overall, a decent font manager, but with one major drawback.

All fonts are added to TypeDNA in place. They are also all activated in place, which I personally prefer. The only time fonts are written somewhere else is if you activate a duplicate in the list. When you choose to activate a conflicting font, that font is copied to the Fonts folder of your user account.

Groups are not easy to manage. After you create a group, you cannot directly add fonts to one by dragging in new fonts from the desktop. Anything you want in a group has to first be part of All Fonts. Then you drag from that list (or Recently Added) into your group to create what is really a sub-group. That is, a link to a link. Needless to say, this makes creating groups for individual projects a real chore.

Let's say, over the course of a couple of weeks or months, you've added a couple thousand fonts to TypeDNA. Since the file name of fonts don't always match the names you see in your font lists, it can be almost impossible to find the name of a font you just added. TypeDNA does give you a Recently Added listing, but among the heading choices at the top of the list, you can only realistically sort by name. By date added would be a tremendous help, but there is no such choice.

How many Recently Added fonts does it list, and for how long? Hard to say. I added 100 fonts over two weeks ago, and they're still all listed. When do they drop off so you know what's been more recently added? Basically, you'd have to wade through your 2,000 font names to try and find the ones you just added so you can copy them to your project's group name. This will quickly become a lesson in frustration when you added a font named JT37.ttf, and the name it produces in the list is *Harry Regular*. How are you supposed to know that so you can look for it?

TypeDNA also uses confusing messages. When you add fonts, it tosses out the message, **Font(s) are already processed**. Which leads you to believe you've already added these fonts in the past, and it's letting you know. Uh, no. TypeDNA means it as, "Wow! We've already processed the fonts you added!" Fortunately, you can turn this meaningless message off.

To see how you may be able to return the favor and confuse the app, I added fonts, activated one and a duplicate, then deleted the fonts from the location I added them from. TypeDNA froze and I had to do a Force Quit. Upon a relaunch, TypeDNA realized the fonts were missing and removed the names from the lists. But the duplicate font I had activated, Cooper Black, was now permanently activated by being stranded in the Fonts folder of my user account. And TypeDNA wasn't even aware of its existence, despite being within its own subfolder.

I normally never keep any fonts in the user account Fonts folder. So, what does TypeDNA do with fonts that already exist in this folder when you first start using TypeDNA? The answer is, it lists any such fonts under All Fonts. If you deactivate any of these preexisting fonts, they remain in the user's Fonts folder, but are correctly deactivated in the system and can be reactivated at any time.

TypeDNA marks any system font that begins with a period, such as .Lucida Grande UI Regular, as being inactive. No, they're active and you can't turn them off. But they're marked as off, regardless.

What happens when you add same-named fonts? For the test, I took a very old TrueType font and created a copy that uses **A** for every upper case letter A-Z, and **a** for every lower case letter a-z. The result here is, it added both to All Fonts. The original correctly listed as its internal name of Avian Regular, and the modified font as Aaaaa Aaaaaa. This simply means yes, you can add same-named fonts. TypeDNA generates the WYSIWYG display of font names you see from the glyphs in each font. If you keep adding an identical font from different locations, they continue to be added to the list; all with the same name. The only way to know which is which is to right click on one of them and choose Show in Finder.

Here, TypeDNA does recognize that you have multiple, conflicting fonts. If you already have one active and try to activate another instance of the same font, you get a popup box warning you of the conflict and lets you choose to activate the requested font, or leave things as they are.

As it functions now, TypeDNA falls under the heading of okay for light, home use. It's a step up from much simpler font managers, but isn't the easiest app to use if you have to routinely manage hundreds, or thousands of fonts. The one thing the makers of this app could do to make it *much* better is to allow you to drop new fonts directly into a group. Then you wouldn't have to wade through a monolithic list of

fonts to add them to a group after the fact.

6) RightFont: Nice looking, but a simple font manager with very few controls.

I don't know about you, but I'm not a fan of forced forefront apps. Unless you click the button to minimize RightFont back to the Dock, it is always on top of everything else.

You can change a preference to allow duplicate fonts, but nothing actually happens when you do that, other than a message appears saying duplicate fonts have been imported.

There are no sets of any kind. When you import fonts, they simply get added to an increasingly longer list. For heavy font users, you will end up with thousands of entries with no sorting other than by name. All fonts you import are automatically copied to a folder by the name *RightFont* in your user account. They are separated alphabetically by subfolders. When you do import a duplicate, there is still, and always, only one of that font. Which explains why the interface gives you no choice of one or the other.

This is bad. As noted in other font manager reviews, you must be able to truly allow access to fonts that have the same names. Clients long have, and still do send in modified fonts. If you use the wrong one on a project, it could cost you hundreds, or thousands of dollars to redo the work.

Does it work? Yes, very well for what it does. Is it a font manager I would recommend to professionals? No. It's a very minimalist manager with nowhere near enough features to be used in a production environment.

How to handle font caches

All types of software create and maintain cache files. The purpose of a cache file is to access frequently used data faster than it can usually be obtained from its original location. Since we're talking about fonts, we'll use them as an example. Any font activation causes the OS to create cached data of those fonts. This data is optimized for both faster retrieval and rendering; much faster than it would be to access the fonts themselves every time a call is made to draw a font to the screen. If over time you use a particular font less, even though it's active, the cached data for that font may eventually be removed from the cache in favor of those being used more frequently. If you use the font again, its data will be added back into the cache in the assumption that the data will once again be needed more often. Hence, cache data is something that is always in flux whether it's font data, data for the OS itself or any other program that use various types of caches.

Because that data is read from and written to so often, the data is bound to have errors introduced at some time. When that happens, you see odd problems. With fonts, it's almost always garbled text caused by corrupt font caches.

Utilities you can use to clear your font caches are [Font Nuke](#), [Cocktail](#) and [OnyX](#). The last two are powerful utilities which can do far more than just cleaning font caches. If you are not familiar with these applications, then I would caution you in their use as you could cause your Mac to become non functional, requiring a reinstall of the OS.

Suitcase and FontExplorer X Pro both come with cache cleaning tools. With Suitcase Fusion in the foreground, the option is under [File > Clean Font Caches](#). This is a very thorough cleaning. It clears the the OS font cache files, and those for many third party apps which maintain their own separate font cache files. A nice addition starting with FontAgent Pro 5, Smasher is now included as part of the package. No need to purchase it separately.

FontExplorer X Pro has its tools broken down into three options:

(1) Clean System Fonts Folder

This choice refers to the entire system, not just the System folder. What this tool does is move any font that is not installed as part of the OS to a new folder on your desktop. A very quick and easy way to separate the OS X supplied fonts from those you've added. (Lion and later duplicate this feature in Font Book). This can be very helpful when you're having trouble tracking down font issues. Once you run this option along with cleaning cache files and the remaining fonts are still misbehaving; then you know you need to at least reinstall the OS supplied fonts. See the instructions at the bottom of this article to see how.

(2) Clean System Font caches

Clears all font cache files for both the system and the active user account.

(3) Clean Application Font caches

Specific to clearing font cache files created by the Adobe apps, Quark XPress and Microsoft Office.

Notes on cache files:

There is a right way and a wrong way to remove cache files. OnyX force you to do it the right way by restarting immediately afterwards. So why is this right? As just noted, cache files are accessed frequently. If you remove any type of cache file and attempt to continue working, you could easily have just pulled the rug out from under an application that was reading (or worse, writing) to a cache file that was removed during the cache cleaning operation. The consequences can be anywhere from minor to disaster. The rules for clearing cache files are simple, but need to be followed to avoid potential data loss.

- 1) Quit all running applications. This is not optional - do it. While it doesn't hurt to leave your font manager running, as long as it's in a static state (not actively being used in any way), the only 'program' running should be the OS itself. If you want to really be certain, shut down your font manager, too.
- 2) Use any of the utilities linked to in this section to remove the font cache files (Suitcase Fusion 3 through 8, and FontExplorer X Pro can also remove font cache data using an option within the application). If the program you use does not automatically force a restart, don't take that to mean it's okay to continue working. Restart your Mac **immediately**. Under no circumstances should you ever skip restarting after manually removing any type of cache data.

Rules do have exceptions, and one is Microsoft Office. It has its own font cache data which can be removed without restarting your computer. Though you still **must** first shut down any running Office applications. After closing all Office apps, remove the following files:

For Office 2011, the location is:

`~/Library/Preferences/Microsoft/Office 2011/Office Font Cache`

The Office 2011 Service Pack 2 (version 14.2.0) changed the location of many preference files, including the font cache data. It is now located at:

`~/Library/Application Support/Microsoft/Office/Preferences/Office 2011/Office Font Cache`

After removing the file, you can then launch any Office application to have it rebuild its font cache data.

Office 2016 does not appear to maintain its own cache. So a restart is required after you clear font cache data.

In OS 10.11.x El Capitan through 10.14.x Mojave, you can also remove font cache files using the Terminal application.

Close all running applications. From an administrator account, open the Terminal app and enter the following command (or copy/paste it from here):

```
sudo atsutil databases -remove
```

Enter your administrator password when prompted.

This removes all font cache files maintained by macOS. Both for the system and the active user account. After running the command, close Terminal and **immediately** restart your Mac.

Font 911

Q: "I removed the San Francisco fonts while cleaning out my system. Now my Mac won't boot. Help!"

A: There are ways to get these critical fonts back onto your Mac.

- 1) This method is much less invasive or time consuming than reinstalling the entire OS. The only drawback is that you need access to another Mac running the same basic version of macOS (El Capitan, Sierra, etc.). Any Mac running El Capitan or later supports Target Disk Mode. Which methods you can use are listed in [this article](#).

Place the non functioning Mac next to a working Mac and connect the two directly to each other using a cable that is common to the method they can use. It doesn't matter if the Macs are on or not. The various connections mentioned in Apple's article are all hot plug-able connections and will not damage anything by doing so, though it is easiest to have both Macs off to start with. If the functioning Mac is not already on, turn it on first and wait until it is in a ready state. If the non working Mac is on, turn it off by pressing and holding the power button until it shuts off. Turn on the non booting Mac. Immediately hold down the T key to put the Mac into Target Disk Mode. Keep

holding the T key until a large Target Disk Mode icon appears on the monitor. The hard drive of that Mac will now appear as a removable drive on the working Mac. Open it as you would any other disk and copy the San Francisco fonts from the working Mac to the same location on the non functioning Mac. The location is `/System/Library/Fonts/`. Once copied, dismount the hard drive of the Mac in Target Disk Mode by highlighting its icon and pressing Command+E. Or drag its icon to the Trash, which will change to an Eject icon. Turn the non working Mac off again using the power button. Disconnect the cable. When you restart your repaired Mac, it should boot.

2) The second method is similar to the first. Except here, you're using an external drive with the same version of macOS on it.

Have your non functioning Mac off. Press and hold the power button if necessary to power the computer down. Connect your external drive to the Mac and turn the external drive on. Wait 10 seconds or so to give the drive time to get to a ready state. Turn on the Mac and immediately hold down the Option key. Continue holding the key until you get a screen showing which drives have a bootable system on them. This will take a minute or so as the Mac tries to determine if there's a network drive attached. Once it is done searching (the normal arrow mouse cursor will appear), select the external drive by double clicking on its icon. The Mac will boot to the external device. Once ready, you can then copy the San Francisco fonts from the external drive to your main drive as described in method one. As the Option key method of booting is a temporary choice, you don't need to open the System Preferences and reselect your main drive as the startup disk. Simply restart and your Mac will boot to the internal drive. Perform a Repair Permissions on your system when it has finished starting up.

3) For all Macs which shipped with Lion or newer, and you have no other drive or partition to boot to, restart the Mac and hold down the Command+R keys to boot to the Recovery partition. Reinstall the OS.

Q: "The fonts in Safari and many other applications are a garbled mess, or Safari and other applications won't launch."

A: A common cause of applications not starting are corrupt fonts. But Helvetica Fractions in particular is known to cause problems, as is Times Phonetic. Don't just disable these fonts in your font manager, but physically remove them from your system and try your applications again.

Q: "What in the world is with a question mark in a box?"

A: Are you seeing a strings of characters like this image where there should be text in your web browser, email or other app?



The most common causes are that your font cache data is damaged, or you've turned on conflicting fonts. When either happens, the OS opts for the last resort to display something, *anything* there. Appropriately, this text is from the system font LastResort. Usually, for corrupt cache data, all you have to do is restart your Mac. If that doesn't do it, then clear the font cache files from the system using Terminal, as described at the bottom of the previous section. If the problem appeared after enabling some new fonts, turn them back off and see if the problem corrects itself. If neither helps, you may need to reinstall the OS.

Yosemite was the last time the boxed A was used. There was quite a variety of boxed glyphs in the LastResort then. In El Capitan and later, Apple threw almost all of them out. The main glyph used to signify a missing font now is a boxed question mark.

Determining if fonts are causing problems with applications

This can be difficult, as the problem could be related to fonts, preference files, system files, or a combination of the three.

First move all fonts in the `/Library/Fonts/` and the `~/Library/Fonts/` folders to a new folder on the desktop. If you have pared your fonts down to the minimal list, this will leave only the basic fonts active in the `/System/Library/Fonts/` folder (see list above in Section 1).

If your applications are still having problems, you then you need to determine if the problem is damaged system fonts or preference files. To do so, open the System Preferences. Click on the Accounts icon and create a new user account. Log in to that account and run the applications you are having trouble with. If they are still not operating, then it is likely that your system fonts are damaged, or possibly the OS.

Your next least intrusive step is reinstalling the fonts using the method below in the section, **Reinstalling your fonts**, or obtaining them from another Mac or external drive using one of the methods above.

If after all this your applications are still not functioning, then other basic OS system files are damaged; which only a reinstall of the OS can repair.

If your applications are now working correctly in a new user account, but you are still experiencing problems:

Then the preference files in your original user account may be what is damaged. There is no easy way to fix this, either. At least not one that will keep you from losing all of your individual preferences for each application. I suggest you open the Preferences folder in your user account and move everything in that folder to a new empty folder on your desktop; so the original is empty. Restart your Mac and login to your normal account. When the Mac is ready, you can remove the new account you made for testing in the first step if you tried that.

At this point, everything about your desktop will look as it did when you first installed the OS. The Dock, your desktop image, everything. For those applications that need certain preferences present to avoid having to reinstall them, you can try copying those particular preferences back from the folder you created on your desktop containing your original preference files and see if they work. If not, that preference file is damaged and you may need to reinstall the application. Others will simply let you reenter the serial number when it needs to create a new preference file.

Why you wouldn't just reinstall the OS? Because your preference files would still be damaged if you reinstalled the OS, preserving your settings. If you performed a clean install, then your preferences would be gone and replaced with new ones. So you either will have accomplished nothing with a reinstall, since your original damaged preference files will still be there; or you will put yourself at the same point you would be if you had simply removed your preference files. Except it took you a lot longer to do the latter because you did a complete clean reinstall of the OS rather than just empty the Preferences folder.

Another option: You can also move all user fonts to one location to determine if only fonts are the problem. Move all fonts in the `/Library/Fonts/` and `~/Library/Fonts/` folders to a new folder on your desktop. If your applications run normally, then one or more of your now disabled fonts is causing the problem. It doesn't matter if the application uses a particular font or not; a damaged font can cause memory leaks and other memory related problems. Copy or move five fonts at a time into the `/Library/Fonts/` folder. If behavior problems return, then one or more of the fonts you reactivated are damaged. Replace as needed.

Reinstalling your fonts

It is possible to restore your fonts without resorting to a reinstall of the entire OS.

Restoring fonts on Macs that shipped with (or have been upgraded) to El Capitan through Mojave:

Since Lion, Macs do not ship with any removable media for restoring the OS. You boot to the emergency partition by restarting and holding down Command+R. From there you can restore the OS or erase a drive and reinstall. A full install package for El Capitan or later is at least 5 GB. The entire hidden recovery partition is only 650MB. That tells you right away that this is not an entire system. If you do an erase and install, as much of the OS as is possible will be installed from the recovery partition. The rest will be downloaded from Apple's servers and installed.

I wanted to see what fonts could be restored from the recovery partition. The result is that they are in a hidden .dmg file. In that disk image is a folder named `/System/Library/Fonts/Base/`. As an example, of the 245 fonts Mountain Lion ships with, only 51 fonts are in this folder. This one folder is actually a combination of fonts which normally go into the `/System/Library/Fonts/` folder, and others in the `/Library/Fonts/` folder. It's not easy to get to. You need to first modify Disk Utility's preferences so it will show hidden partitions so you can mount the Recovery drive. Then you run a Terminal command to show hidden files and folders. It's not something a novice should attempt, so I'm not going to explain the entire process. There are so few fonts to recover from there anyway it isn't worth the effort.

In High Sierra and later, the Recovery partition is not something you can even get at. It's integrated into the same partition the OS is on.

Whenever you get the opportunity to download an entire El Capitan or later system installer through the App Store (as in a system upgrade), do so. When the download finishes, it will want to install the OS. Immediately press Command+Q to quit the installer. You can then backup the full installer to a flash drive or external hard drive. Then you have it and can use the following instructions.

The following instructions assume you copied your El Capitan through Mojave download to an external drive or other location to save for future use.

Font locations for El Capitan through Mojave:

First, download [Pacifist](#). After Pacifist has finished reading all of the installer packages, select the package that has the fonts you want installed (see below) and click the Install button on Pacifist's menu. Only that package will be installed to its default location. For a complete set of fonts, reinstall both the `/System/Library/Fonts/` and `/Library/Fonts/` folders. All fonts you install from Pacifist will overwrite any identical fonts on your hard drive in the default locations, replacing any damaged fonts in the process. If you have copied or moved any of the default fonts to another location, they will still be there and will need to be deleted manually. All in all, this method is much quicker and easier than doing an entire reinstall of macOS to get your fonts back where they belong.

Since the OS that shipped with a Mac is usually machine specific, you can use the instructions below to reinstall the fonts, but do not use the .dmg from the App Store to reinstall the OS. To do that, boot to the Recovery partition (Command+R startup), or Internet Recovery (Command+Option+R startup).

Right click on the OS installer and choose **Show Package Contents**. Double click the **Contents** folder, then the **SharedSupport** folder. Double click the file **InstallESD.dmg** to open the disk image (this step is not necessary for El Capitan). A disk image drive icon will appear on the desktop. A window should automatically open when the disk image is mounted. But if not, double click the desktop icon to open it. Double click the **Packages** folder.

Open the file **OSInstall.mpkg** with Pacifist. You can do this by dragging and dropping the .pkg file onto Pacifist's interface. Once you have navigated to the Fonts folder of any location shown below, highlight the folder *Fonts* by clicking on it once. Click *Install* at the upper left of the Pacifist window. This will reinstall all fonts within that folder. You can also expand the Fonts folder to highlight individual fonts to reinstall only those you choose.

El Capitan System fonts: [Contents of EssentialSystemSoftware > Contents of EssentialSystemSoftwareGroup > Contents of Essentials > System > Library > Fonts](#)

El Capitan Library fonts: [Contents of EssentialSystemSoftware > Contents of EssentialSystemSoftwareGroup > Contents of Essentials > Library > Fonts](#)

Sierra System fonts: [Contents of OSInstall > Contents of Essentials > System > Library > Fonts](#)

Sierra Library fonts: [Contents of OSInstall > Contents of Essentials > Library > Fonts](#)

High Sierra and Mojave System fonts: [Contents of OSInstall > Contents of Core > System > Library > Fonts](#)

High Sierra and Mojave Library fonts: [Contents of OSInstall > Contents of Core > Library > Fonts](#)