

Font Management in macOS and OS X, by Kurt Lang

Last updated July 9, 2017

This article deals with font usage in OS X Panther 10.3.x through macOS Sierra 10.12.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.

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?

- Review of the font manager, TypeDNA.

Table of contents

Preface: Understanding terminology used in this article.

Sections:

1. Required fonts.
2. Known issues with disabled fonts.
3. Fonts installed by Microsoft Office.
4. What to do with Suitcase.
5. Controlling Helvetica and Courier fonts for prepress.
6. Font folder locations. Using Font Book.
7. Required fonts for OS 9 (Classic).
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 and OS X.
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 or OS X 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 Mac OS releases (Panther 10.3 through Sierra 10.12) 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 OS X 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, Keyboard.dfont, LastResort.dfont and LucidaGrande.dfont are used mainly for menus and other system font display purposes; therefore, they are the most important to the OS itself. In Mountain Lion and earlier, you must never remove Lucida Grande. Without that font, the system will not boot. If you remove it while the system is active, you will lose control of all menus (they will be blank), essentially locking you out of your Mac. Mavericks utilized a different method to protect access to the desktop (see the specific text alongside the minimal font list for Mavericks). In Yosemite, Lucida Grande is no longer the main system font, and HelveticaNeueDeskInterface.ttc is the font you cannot remove for any reason. El Capitan and Sierra change things again with the introduction of an all new set of system fonts; San Francisco. This set, HelveticaNeueDeskInterface.ttc and LucidaGrande.ttc must be present for the Finder 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](#), [MasterJuggler](#), [FontExplorer X Pro](#), [TypeDNA](#), or other font manager. Note that MasterJuggler is PowerPC only, and so is suitable for use only in Snow Leopard 10.6 or older versions of the Mac OS.

Beginning with Leopard, 10.5 and up through Mountain Lion 10.8, Apple made it difficult to remove critical fonts. If you attempt to remove protected fonts from the /System/Library/Fonts/ folder, the OS will tell you that you cannot remove the font(s) and immediately replaces them from copies in another location. There are many fonts you can still remove from the /System/Library/Fonts/ folder, but some will resurrect themselves. See section 5 on how to permanently remove Apple's supplied versions of Helvetica and Helvetica Neue in Leopard, 10.5 through El Capitan, 10.11 if this is important for you. Mavericks 10.9 and later eliminated this type of font protection, but you still need to remove the supplied fonts if they conflict with other types of fonts you are required to use. El Capitan 10.11 and Sierra 10.12 add 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 and OS X 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 and OS X users should have in their `/System/Library/Fonts/` folder, rather than the leaning towards the needs of prepress. So 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, even though they are not listed as having a period as part of the name by the OS. Not even if you do a file listing in Terminal. 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.

Apple has tied their conflicting versions of Helvetica so closely to the OS, and in so many places, it is no longer easy to manage them so you can use a Type 1 PostScript version. 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 5 if this applies to you).

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

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, Bold Condensed, UltraLight, UltraLight Italic, Condensed Black, Condensed Bold

With El Capitan, Apple has (almost) released Helvetica and Helvetica Neue back to the user. Really! Bad news. Adobe, Microsoft and possibly other third party vendors have not. See the minimum font list for El Capitan for more information.

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 2 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/HelvelTMM  
/System/Library/Fonts/Helvetica LT MM  
/System/Library/Fonts/Times LT MM  
/System/Library/Fonts/TimesLTMM
```

The message is wrong since a default install of Lion through Sierra *will* install these files (actually, El Capitan and Sierra will only install two of them). Font Book runs the Unix command `fontrestore` under the option **Restore Standard Fonts**. When run, it does indeed remove all of 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 Lion 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 or OS X 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. So while some will come back, most won't. The command also removes fonts which are not part of the macOS or OS X 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 or OS X. The active user account Fonts folder gets emptied out.

Starting with Lion, 10.7.x, Apple made the decision to hide the Library folder in the user accounts. Most tech writers presume this was to make it more difficult to locate and delete files a person shouldn't be digging through (without knowing what they're for). In order to see the Library folder of your account, Apple's method involves (1) being at the desktop, (2) hold down the Option key and (3) choose [Go > Library](#) from the menu. For advanced users, this gets really old since it's surprising how often you do go into your Library folder for various reasons. To make it permanently visible, open Terminal and enter the following command:

```
chflags nohidden ~/Library
```

This will make the Library folder visible for the account you are logged into at the time, so the command will need to be repeated for each user account as you log into each one. This change will also be reset if you reinstall the OS or apply any updates. You will then need to repeat the process.

In Mavericks and later, Apple has made showing your home Library folder much easier. 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+U. 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 or OS X 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 or OS X 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 2015 apps will not launch if Helvetica is missing. Such testing is sometimes more involved than that. For instance, the Microsoft Office 2016 apps will not reliably launch unless HelveticaNeue.dfont is specifically in the [/System/Library/Fonts/](#) folder. There is an easy way around this issue for prepress and other professionals who rely on the use of Type 1 PostScript versions of Helvetica Neue. See section 5 for details.

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 6).



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.

Use the same general instructions for El Capitan to remove fonts from the [/System/Library/Fonts/](#) folder in Sierra. One big change is that once you disable System Integrity Protection, you don't have to boot to a different drive to remove fonts from the System folder. Not only that, you don't even need to restart your Mac! Put them in the trash and empty it. Re-enable SIP when you've finished removing the fonts.

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
SFCompactDisplay-Black.otf
SFCompactDisplay-Bold.otf
SFCompactDisplay-Heavy.otf
SFCompactDisplay-Light.otf
SFCompactDisplay-Medium.otf
SFCompactDisplay-Regular.otf
SFCompactDisplay-Semibold.otf
SFCompactDisplay-Thin.otf
SFCompactDisplay-Ultralight.otf
SFCompactRounded-Black.otf
SFCompactRounded-Bold.otf
SFCompactRounded-Heavy.otf
SFCompactRounded-Light.otf
SFCompactRounded-Medium.otf
SFCompactRounded-Regular.otf
SFCompactRounded-Semibold.otf
SFCompactRounded-Thin.otf
SFCompactRounded-Ultralight.otf
SFCompactText-Bold.otf
SFCompactText-BoldItalic.otf
SFCompactText-Heavy.otf
SFCompactText-HeavyItalic.otf
SFCompactText-Light.otf
SFCompactText-LightItalic.otf
SFCompactText-Medium.otf
SFCompactText-MediumItalic.otf
SFCompactText-Regular.otf
SFCompactText-RegularItalic.otf
SFCompactText-Semibold.otf
SFCompactText-SemiboldItalic.otf
SFNSDisplay.ttf
SFNSDisplayCondensed-Black.otf
SFNSDisplayCondensed-Bold.otf
SFNSDisplayCondensed-Heavy.otf
SFNSDisplayCondensed-Light.otf
SFNSDisplayCondensed-Medium.otf

SFNSTextCondensed-Regular.otf
SFNSDisplayCondensed-Semibold.otf
SFNSDisplayCondensed-Thin.otf
SFNSDisplayCondensed-Ultralight.otf
SFNSText.ttf
SFNSTextCondensed-Bold.otf
SFNSTextCondensed-Heavy.otf
SFNSTextCondensed-Light.otf
SFNSTextCondensed-Medium.otf
SFNSTextCondensed-Regular.otf
SFNSTextCondensed-Semibold.otf
SFNSTextItalic.ttf
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.



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 5 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 OS X 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 still rather continue on with method two, here's how. Restart and hold down the Command+R keys to boot into Recovery Mode. You can also start up to a flash drive you have created a bootable full installer of El Capitan on. Launch Terminal from the Utilities menu. Enter the command:

```
csrutil disable
```

Close Terminal and restart normally. 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.

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

Do not re-enable System Integrity Protection yet. You can't empty the trash 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. So 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. Only this time in Terminal, enter:

```
csrutil enable
```

Then once again restart normally.

Sierra is due for its official release in just a couple of weeks as I write this, but an issue regarding Messages just came to light. 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.ttc
SFNSDisplay-Black.otf
SFNSDisplay-Bold.otf
SFNSDisplay-Heavy.otf
SFNSDisplay-Light.otf
SFNSDisplay-Medium.otf
SFNSDisplay-Regular.otf
SFNSDisplay-Semibold.otf
SFNSDisplay-Thin.otf
SFNSDisplay-Ultralight.otf
SFNSText-Bold.otf
SFNSText-BoldG1.otf
SFNSText-BoldG2.otf
SFNSText-BoldG3.otf
SFNSText-BoldItalic.otf
SFNSText-BoldItalicG1.otf
SFNSText-BoldItalicG2.otf
SFNSText-BoldItalicG3.otf
SFNSText-Heavy.otf
SFNSText-HeavyItalic.otf
SFNSText-Light.otf
SFNSText-LightItalic.otf
SFNSText-Medium.otf
SFNSText-MediumItalic.otf
SFNSText-Regular.otf
SFNSText-RegularG1.otf
SFNSText-RegularG2.otf
SFNSText-RegularG3.otf
SFNSText-RegularItalic.otf
SFNSText-RegularItalicG1.otf
SFNSText-RegularItalicG2.otf
SFNSText-RegularItalicG3.otf
SFNSText-Semibold.otf
SFNSText-SemiboldItalic.otf
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.



OS X, 10.10 Yosemite

Lucida Grande has lost its *King of the Hill* status. We now know where Apple was headed with the font HelveticaNeueDeskUI.ttc which first appeared in Lion, but had no apparent purpose. The new system font in Yosemite is HelveticaNeueDeskInterface.ttc.

HelveticaNeueDeskInterface.ttc is critical to Yosemite. If you remove it, there is only a partial fall back to another font in the System folder. As long as Helvetica.dfont or HelveticaNeue.dfont are still available, you will at least get an active desktop with text you can read. However, all menu items will drop down a blank menu. If you're in this predicament, you can press the power button to cause a menu to appear which will allow you to restart.

If you remove all three Helvetica fonts, you will completely lose control of your Mac. The OS will start flashing between the desktop and a blank screen. Any text that does appear will use the LastResort font and will be unreadable. You can't click on anything, or get out of it other than a forced shut down by holding the power button.

If you make either mistake, you must either be able to boot to another partition and copy at least HelveticaNeueDeskInterface.ttc back into the `/System/Library/Fonts/` folder from a backup, or reinstall the OS.

It's possible in the future that a font dedicated to the desktop is good news for those who must deal with the conflict issues of Type 1 PostScript versions of Helvetica and Helvetica Neue with Apple's supplied versions. But for now, many of Yosemite's apps still require Lucida Grande, and/or Apple's Helvetica or Helvetica Neue fonts to launch.

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

Apple Color Emoji.ttf
AppleSDGothicNeo-Bold.otf
AppleSDGothicNeo-Regular.otf
Courier.dfont
Geneva.dfont
Helvetica.dfont
HelveticaNeue.dfont
HelveticaNeueDeskInterface.ttc
Keyboard.ttf
LastResort.ttf
LucidaGrande.ttc
Menlo.ttc
Monaco.dfont
Symbol.ttf
Times.dfont

Fonts updated in 10.10.3

Apple Color Emoji.ttf



OS X, 10.9 Mavericks

Apple has shifted fonts around again in Mavericks. The Avenir, Palatino and Optima fonts, which were in the `/Library/Fonts/` folder in Mountain Lion are now in the `/System/Library/Fonts/` folder. The system Thonburi .ttf fonts have been combined into a single .ttc font.

Apple Gothic no longer appeared to be a required font, but a change in 10.9.2 has brought it back. The Messages app now fails if this font is missing. The default font for Notes is now Helvetica, so MarkerFelt is also no longer needed. You can easily change the default font for Notes back to MarkerFelt if you wish. Leave MarkerFelt on your system if you prefer to use that font.

The keyboard viewer has changed in Mavericks to use the font AppleSDGothicNeo. It will display very strangely if this font is not active.

Be extremely careful when removing fonts in Mavericks. There is no longer a ProtectedFonts folder. If you remove a critical system font and delete it, it's gone and will need to be replaced from a backup, or by reinstalling the OS if you have no backup.

In all previous versions of OS X, the loss of Lucida Grande would make your system unusable. What Mavericks does differently as a form of font protection is to use the first available system font (not including Apple Color Emoji) for the menus. I had already minimized the fonts in my System folder, so when I removed Lucida Grande, Mavericks switched to Courier, which was next in the list. If I removed that, the menus switched to Geneva. Mavericks will continue to move down the list of available fonts alphabetically.

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

- Apple Color Emoji.ttf
- AppleSDGothicNeo-Bold.otf
- AppleSDGothicNeo-Regular.otf
- Courier.dfont
- Geneva.dfont
- Helvetica.dfont
- HelveticaNeue.dfont
- HelveticaNeueDeskUI.ttc
- Keyboard.ttf
- LastResort.ttf
- LucidaGrande.ttc
- Menlo.ttc
- Monaco.dfont
- Symbol.ttf
- Times.dfont

Fonts updated in 10.9.2

Apple Color Emoji.ttf, AmericanTypewriter.ttc, ArialHB.ttc



OS X, 10.8 Mountain Lion

See the font list under Lion, 10.7. The minimum font list for 10.8 is *almost* identical to 10.7. AppleGothic.ttf is in the `/Library/Fonts/` folder in Mountain Lion. It likely wasn't intended to be, but that's where it is. Be sure to keep AppleGothic.ttf when doing any purging of the `/Library/Fonts/` folder. Otherwise, all other fonts are the same in Mountain Lion for the `/System/Library/Fonts/` folder as they are in Lion.

The only other semi font related difference is that Notes has been separated from Mail as an individual app. As in previous versions of OS X, the default font for Notes is MarkerFelt.ttc.



OS X, 10.7 Lion

At first glance, it appeared Lion had eliminated Helvetica Light and Helvetica Light Italic from its required fonts. They are still there and have been moved into the Helvetica.dfont package. HelveticaNeue, which was a .ttc font package in Snow Leopard, is now once again a .dfont. So .dfonts aren't as obsolete as it once appeared Apple was going to make them.

MarkerFelt.ttc has apparently been forgiven of some transgression by the rest of the fonts in the `/System/Library/Fonts/` folder and has

returned there from its banishment in the [/Library/Fonts/](#) folder of Snow Leopard. If you use an email program other than Mail, you can remove Markerfelt.

I really don't know why HelveticaNeueDeskUI.ttc is a protected font. From the name, one would guess that DeskUI stands for **Desktop User Interface**. I've removed it from the system entirely in testing, and Lion didn't seem to care in the least that it was gone. It's also one of those fonts that Suitcase displays beginning with a period. In fact, even the names of the four fonts within the TrueType Collection start with periods. Because of that, even when active, they don't show up in any application. They're not hurting anything to be in the System folder, and Apple must have *some* purpose for them; I just can't see what that may be.

While you may never use it, Apple Color Emoji.ttf must be left in the System's Fonts folder. It will only cause a problem if you open the Character palette and click on the Emoji heading, but then the Character palette will be royally stuck.

Previously, you could remove Apple's Helvetica fonts and permanently use Type 1 PostScript versions in their place. Not any more. Address Book is one app which will not launch if Apple's version of Helvetica is missing. So for prepress, designers or others who must use Type 1 PostScript versions of Helvetica or Helvetica Neue, it is now a requirement to copy Apple's Helvetica fonts to a non Fonts folder location, remove them from the System and ProtectedFonts folders, and then set up separate sets in your font manager for Apple's Helvetica fonts, and another for your third party Helvetica fonts. Then turn on whichever set you need at the time and turn the other set off. It's not perfect as you may need the the Type 1 fonts open, which will leave you temporarily unable to open Address Book until you disable the Type 1 fonts and turn the Apple supplied versions back on. There is no way around it. You simply can't have both on at the same time.

So other than this new problem with Helvetica for those who need the use of Type 1 PostScript Helvetica fonts, the minimum System folder font list for OS X, 10.7 is:

- Apple Color Emoji.ttf
- AppleGothic.ttf
- Courier.dfont
- Geneva.dfont
- Helvetica.dfont
- HelveticaNeue.dfont
- HelveticaNeueDeskUI.ttc
- Keyboard.ttf
- LastResort.ttf
- LucidaGrande.ttc
- MarkerFelt.ttc
- Menlo.ttc
- Monaco.dfont
- Symbol.ttf
- Times.dfont

A handful of fonts have received an update for Lion over the course of the OS updates noted next. If you had previously removed the updated fonts from the [/Library/Fonts/](#) folder, the [/System/Library/Fonts/](#) folder and the [ProtectedFonts](#) folder, you will need to do so again. You can see some are repeated as some of the fonts have been updated more than once. Also, if you commonly use the Combo updates, they will contain all fonts which have been updated since 10.7.0. So even if there are no font updates in a given point release, you will still be reinstalling all fonts which have received changes. There are some minor date differences between 10.7.3 and 10.7.4 on a few fonts, but only by a few days. I would suspect they weren't actually changed.

Fonts updated in 10.7.2

In the [/Library/Fonts/](#) folder:

- Devanagari Sangam MN.ttc
- Myanmar MN.ttc
- Myanmar Sangam MN.ttf

In the [/System/Library/Fonts/](#) and [ProtectedFonts](#) folders:

- HelveticaNeue.dfont
- HelveticaNeueDeskUI.ttc

Fonts updated in 10.7.3

In the [/Library/Fonts/](#) folder:

Ayuthaya.ttf
Devanagari Sangam MN.ttc
Krungthep.ttf
Myanmar MN.ttc
Myanmar Sangam MN.ttc
Sathu.ttf
Silom.ttf

In the `/System/Library/Fonts/` and `ProtectedFonts` folders:

HelveticaNeue.dfont
HelveticaNeueDeskUI.ttc



OS X, 10.6 Snow Leopard

The font formats have changed in Snow Leopard, though the main list of fonts shows little change. Apple has replaced many of their proprietary .dfont fonts with standard TrueType OpenType fonts, which have a .ttf or .ttc extension. There are a handful of .dfonts left in the `/System/Library/Fonts/` folder, but for the most part, they're obsolete (see section 8 for additional fonts Snow Leopard installs). As in OS X 10.5, the Multiple Master fonts are visible, but are blank entries which can be removed.

The recommended minimum font list for OS X, 10.6 is:

AppleGothic.ttf
Courier.dfont
Geneva.dfont
Helvetica.dfont
HelveticaLight.ttf (added in 10.6.5)
HelveticaLightItalic.ttf (added in 10.6.5)
HelveticaNeue.ttc
Keyboard.ttf
LastResort.ttf
LucidaGrande.ttc
Menlo.ttc (the new default font for Terminal)
Monaco.dfont
Times.dfont
Symbol.ttf

MarkerFelt.ttc is still required for the Notes portion of Mail in Snow Leopard, but is no longer in the `/System/Library/Fonts/` folder. Instead, it's been moved to the `/Library/Fonts/` folder.

OS X, 10.6.5 Snow Leopard

Two new fonts have been added in this midpoint update. Both of which are added to the `/System/Library/Fonts/` folder, and the `ProtectedFonts` folder (reference section 5). As with Apple's versions of Helvetica and Helvetica Neue, the new HelveticaLight.ttf font conflicts with the Adobe Type 1 version. So you have yet another font to manage or remove in order to use your preferred, or required version of Helvetica Light. Other fonts have received updates, so if you've removed them following your initial install of Snow Leopard, you'll have to remove the fonts again following the install of any Snow Leopard Combo or Delta update which includes them. These fonts are:

In the `/Library/Fonts/` folder:

Hoefler Text.ttc
STHeiti Medium.ttc

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

HelveticaLight.ttf
HelveticaLightItalic.ttf
Menlo.ttc
HelveticaNeue.dfont
STHeiti Light.ttc

In the [ProtectedFonts](#) folder:

HelveticaLight.ttf
HelveticaLightItalic.ttf
HelveticaNeue.dfont



OS X, 10.5 Leopard

In this version, the Multiple Master fonts used by Preview are visible, whereas they are hidden in earlier versions of OS X. Helvetica and Helvetica Neue are now in the [/System/Library/Fonts/](#) folder rather than the [/Library/Fonts/](#) folder. Courier is no longer a critical system font in Leopard. Also notable is that Leopard comes with OpenType versions of the same named fonts installed by Microsoft Office listed in section three. The following list is the minimum set of fonts for OS X, 10.5.

AppleGothic.ttf
Courier.dfont
Geneva.dfont
HelveLTMM
Helvetica LT MM
Helvetica.dfont
HelveticaNeue.dfont
Keyboard.dfont
LastResort.dfont
LucidaGrande.dfont
MarkerFelt.dfont (for the Notes portion of Mail)
Monaco.dfont
Times LT MM
TimesLTMM
Times.dfont
Symbol.dfont



OS X, 10.3 Panther, and



OS X, 10.4 Tiger

AppleGothic.dfont
Courier.dfont
Geneva.dfont
Helvetica.dfont
Keyboard.dfont
LastResort.dfont
LucidaGrande.dfont
Monaco.dfont
Times.dfont
Symbol.dfont

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 Leopard, 10.5 through Sierra, 10.12.

Panther 10.3, and Tiger 10.4 have most of the following fonts as legacy Mac TrueType. Their names will be the same as the shorter list noted at the beginning of section three. Panther, 10.3 does not include Impact. Neither 10.3, or 10.4 include any version of Wingdings.

Recommended minimum fonts for the [/Library/Fonts/](#) folder:

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

Note: In addition to the above list, if you are running Snow Leopard, you must keep MarkerFelt.ttc.

Note: In addition to the above list, if you are running Mountain Lion, Mavericks, Yosemite, El Capitan or Sierra, you must keep AppleGothic.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. Most of these fonts in Leopard, 10.5 and earlier are .dfonts. Also in 10.5 and earlier, Chalkboard and Chalkboard Bold are separate fonts. In 10.6 and later, they are combined in the TrueType Collection, Chalkboard.ttc.

AmericanTypewriter.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 has been moved to the [/System/Library/Fonts/](#) folder in Yosemite, El Capitan and Sierra.

If you use Microsoft Office: See Section 3 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 6 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 or OS X 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 6 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.

Known issues with disabled fonts

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

Courier

Some applications require the presence of Courier in Panther and Tiger.

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 OS X 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.

Helvetica Neue

The Office 2016 apps will not launch if Helvetica Neue is missing.

There's no good reason to remove the macOS or OS X supplied versions 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 5). If you don't, leave them be.

The use of Helvetica has changed starting with Leopard 10.5. While the system normally prevents you from removing either Helvetica or Helvetica Neue, I have removed both from the hard drive and found that iChat and iCal no longer seem to be affected. Both launched and displayed without either Helvetica font available. iPhoto 7 and iLife 08 both require the presence of Helvetica Neue.

In older versions of OS X, Mail is one application that seems to require, or at least prefer the Unicode versions of Helvetica supplied with Leopard. It will work without them as long as you have another version of Helvetica and Helvetica Neue open, but may have some display problems such as text being misaligned or overrunning its intended space. 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.

Monaco

Terminal is one application that will launch, but not display correctly if Monaco is missing (Menlo in Snow Leopard or later).

Times

If you are using the Monaco (X-Rite) GamutWorks application, it requires Times to function. This can be the Times.dfont supplied with OS X, 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.

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

If the font AppleGothic.dfont is not installed, the following problems occur:

- a) You will find that the Adobe CS line of applications will not install in Leopard or Tiger (the CS2 and later applications do not appear to be affected). When trying to install them, you only get as far as entering your administrative password, and choosing your language. The installer then quits. The CS applications run fine once installed. It is only the installers themselves that are affected by the absence of AppleGothic.dfont.
- b) The retail version of OS X Tiger itself has an issue if AppleGothic.dfont is not available. When the DVD is inserted, the disk begins to automatically open, but the Finder then closes the DVD file window and resets before even seeing the contents.
- c) In Microsoft Office 2004 for Mac OS X, previous to version 11.2, PowerPoint will complain that four of the Asian fonts are missing if you are using the short list of fonts mentioned in section one. One error message for each of the four fonts that you need to dismiss. It doesn't effect the operations of PowerPoint, just an annoyance that it insists on telling you that fonts you don't need (as an English speaking user) are missing. You can eliminate this problem by updating to 11.2.x (11.2.1 being current at the time of writing). If you have already attempted to run PowerPoint after updating to 11.2.x without AppleGothic.dfont active, then you have already discovered that PowerPoint crashes after clicking the Open button at the Project Gallery. To fix this problem, activate AppleGothic.dfont. Run PowerPoint again. This time, after clicking Open in the Project Gallery, you will get a message that certain Asian fonts are missing. Turn on the check box at the lower left to tell PowerPoint to never check for those fonts. PowerPoint will thereafter start without errors, or display any messages about missing fonts. Such issues have been eliminated in Office 2008 and 2011. All of its applications will open without any error messages.
- d) In later versions of the Mac OS, the Messages app will not launch if Apple Gothic is missing.

Apple Gothic isn't needed often, but given its importance to viewing and using installation disks and its various effects on both third party and macOS and OS X supplied apps, it is advised to have it enabled at all times.

Fonts installed by Microsoft Office

This section is headlined for Microsoft Office, but also revisits web fonts to note the difference between older legacy Mac TrueType fonts and OpenType versions installed by the various versions of Office.

There are a few fonts installed by OS X and Microsoft programs (namely, Office products) that should be left active since the Microsoft applications (Office X, 2004 and 2008 versions) use these fonts. Mainly for templates. Also, many web sites use these fonts and will display better if they are available on your system. None of these fonts should interfere with any prepress operation in the form of conflicting with a PostScript font of the same name, so can be safely left as is. These are the names as they will appear prior to Leopard, 10.5:

Arial
Arial Black
Arial Narrow
Comic Sans MS
Georgia
Impact
Tahoma

Times New Roman
Trebuchet MS
Verdana
Webdings (from OS 9 or X)
Wingdings
Wingdings 2
Wingdings 3

Note about Wingdings:

OS 9 included Wingdings, Wingdings 2 and Wingdings 3, but OS X did not. It wasn't until Leopard, 10.5 that Wingdings appears as part of OS X. Before Leopard, it was assumed that you could use Wingdings if you installed Classic (OS 9) on your system.

Expanded font list for Leopard through Sierra:

Because the fonts in the `/Library/Fonts/` folder supplied with Leopard through Sierra are OpenType (some are .dfonts) rather than legacy TrueType suitcase fonts, the list *appears* longer. OpenType fonts are saved as one font per file (those with a .ttf or .otf extension). For example, while the Arial legacy suitcase font from Office 2004 looks like one item, it actually contains four fonts; Arial, Arial Bold, Arial Italic and Arial Bold Italic. The OpenType fonts appear as four individual items. The font list for Leopard through Sierra will appear as follows to match the visually shorter list above:

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

Important notes! Leopard, 10.5 through Sierra, 10.12 include many of the same named fonts Office installs on your system. These are larger, OpenType TrueType fonts. You can quickly identify the difference between them a few ways.

- 1) The Apple supplied fonts are larger in size.
- 2) The OpenType fonts have a newer creation date.
- 3) The Apple OpenType fonts have a .ttf, .otf or .ttc extension, the fonts installed by Office X, 2004 and 2008 do not.
- 4) In list view, open the folder window wide enough to see the Kind column. The OpenType fonts will be named as such. Your system may

also denote them as TrueType fonts, but they are OpenType structured fonts.

Because the older fonts from Office X, 2004 and 2008 do not have extensions (they are still legacy OS 9 style suitcase fonts), both end up being installed (normally) in the `/Library/Fonts/` folder, **thus creating a large number of font conflicts**. This doesn't mean you can eliminate font conflicts if the fonts are not in the same folder. Fonts can be spread out in any order, in any active fonts folder, and still create conflicts. See Section 6 for how to find active Fonts folder locations.

To correct these conflicts, you will have to manually remove one instance of each of the same font from the folder or folders in which the conflicting fonts reside. The names or file extensions may not be identical, but if it is the same font, they will create conflicts.

You can resolve the font conflicts in Font Book, but be careful how you do it (see the review on Font Book). Office can use either type of font, so always keep the OpenType fonts. They have far more glyphs (like ligatures, swash characters and other styles) than the older 8 bit, 256 character fonts do not. So for example, keep the fonts **Tahoma.ttf** and **Tahoma Bold.ttf** installed by Leopard through Sierra and remove the older Office font **Tahoma**. The newer OpenType fonts will also provide correct web site viewing with the font names a web page calls for, so there is no need to keep the older versions. There's more to conflicting font names than the names of the files themselves. This is explained in detail later in this article.

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 6).

Office X:

Not very wisely, Office X installed all of its fonts into the `/System/Library/Fonts/` folder. You had to know which fonts belonged to Office and be logged in as an administrator before removing them. Thankfully, Microsoft changed its practice after this version of Office.

Office 2004:

The Office 2004 products install their fonts in the `~/Library/Fonts/` folder (your user account). This is done the first time you run any of the Office products after the initial install in each user account. Tahoma, Trebuchet MS and Verdana are used by Office applications for menus and other program display purposes. All other fonts installed by Office can be removed, leaving only those fonts listed above. They will not be reinstalled by Office after you have removed them. So again, first copy them to another location for future use.

That said, others report that all fonts went to the root `/Library/Fonts/` folder. There seems to be no rhyme or reason to the location the Office fonts end up in. Your best bet is to first examine each of the Fonts folders (see section 6 for locations) before installing Office to see what fonts are already on your system. Then after Office installs its fonts, reexamine the folders to see what has changed and then move, or remove the fonts you don't want active.

Even after being removed, some users have experienced the problem of those removed fonts being reinstalled by Office each time they launch an Office application. One user offered this simple trick to keep Office from reinstalling the fonts: rename the folder where the fonts reside in Office so it can't find them. The location (for Office 2004. The name for Office X will be slightly different) is `/Applications/Microsoft Office 2004/Office/Fonts/`. Simply rename the folder "Fonts" within the Office application folder to something else. Just adding a letter to the beginning should be enough, such as xFonts. **Note:** Be aware that once you rename the fonts folder within the Office folder, if you then create a new user account, none of the fonts will be installed for that user. The easiest was around that though is to take one set of the fonts and put them in the `/Library/Fonts/` folder. Since every user can access fonts in that folder, you can then remove the same named fonts from every user account. Which you actually must do to avoid font conflicts.

Office 2008:

Microsoft did a very nice thing with Office 2008. It installs its fonts only once during the initial installation to the `/Library/Fonts/Microsoft/` folder. No duplicates of the fonts are kept in the Office application folder, nor are they copied repeatedly for each user. Removing them is a snap. Just move the entire **Microsoft** folder out of the `/Library/Fonts/` folder. Office 2008 does not require a single font it comes with to operate, instead using OS X system fonts for its menus, palettes and templates.

If you are running any version of OS X from Leopard, 10.5 through Sierra 10.12, you will still want to prefer to use the same named fonts that come with the OS. They are larger sets than those that come with Office 2008. The fonts from Office 2008, while high quality, are still legacy TrueType suitcase style fonts such as from OS 9 and earlier.

Office 2011:

As with Office 2008, the fonts installed by Office 2011 go to the `/Library/Fonts/Microsoft/` folder. Therefore, as before, it creates conflicts with the existing similar fonts installed by macOS and OS X. If you already have the Office 2008 set of fonts on your hard drive, Office 2011 will replace them. There are also 27 more fonts than in Office 2008. 62 Mac legacy TrueType fonts installed by Office 2011 are identical to those from Office 2008. Six typefaces which used to be legacy fonts are now .ttf fonts. Usually such changes are to fix minor flaws in the

fonts, or to improve kerning and kerning pairs for a more pleasing look.

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 OS X Leopard, Snow Leopard and Lion. The older 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 newer versions have been installed. You can delete the entire [Fonts Disabled](#) folder if you wish.

The .ttc and .ttf fonts installed by Mountain Lion 10.8 through Yosemite 10.10 are identical to those installed by Office 2011, so it doesn't matter which ones you keep. Some in El Capitan and Sierra are newer.

Office 2011 also installs conflicting fonts which are *older* than those supplied with Leopard, 10.5 through Sierra 10.12. These are Mac legacy TrueType suitcase fonts, as opposed to much newer .ttf versions supplied with Leopard through Sierra. So keep the .ttf fonts supplied with macOS or OS X 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:

There is both good and bad about this release. Mostly good. Some of the updates cause radical changes regarding how to stop Office 2016 from showing fonts that don't exist in its lists, among some other things. Since this part was getting a long run-on look, I've divided it by point release so it isn't quite as clumsy to read through. But be sure to read all of the following before proceeding. Which .plist files you remove changes per release. 15.22 and later requires more steps.

With one critical exception, all necessary fonts Office 2016 uses are now embedded within each application package. The only macOS or OS X font they rely on is HelveticaNeue.dfont. Without it, they won't even run.

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 5. If you are seeing this issue, this bug in El Capitan has been fixed as of 10.11.3. Update the OS as necessary.

The issue with fonts that don't exist appearing in the lists of available fonts has been fixed in the 15.17 updates, though you have to manually remove items from the application packages to do it. See below.

15.15

As of the 15.15 update to Office 2016, you can now clear out many items you couldn't before. If you use an app such as Monolingual to remove unused language files on your Mac, you can now use it on Office 2016 without having the apps crash. That's 980 MB of regained space. All fonts in the application packages can also now be safely removed. Since the default in Word, Excel and PowerPoint is Calibri, you should leave that set in each app. Not because they won't work without it, but they will then at least display a font that exists at startup. Removing the large duplication of fonts and excess language files opens up about 2.8 GB of disk space.

15.17

With the 15.17 updates, it is much easier to remove the unnecessary fonts. In each of the five application packages, right click on an app and choose to **Show Package Contents**. In each app, drill down to **Contents/Resources**. Put the folder **DFonts** in the trash. That's all there is to it. Those few fonts Microsoft considers essential to the Office 2016 apps are in the **Fonts** folder within each app. Before proceeding, see the notes below regarding version 15.22 of Office 2016.

The 15.17 updates for Office didn't just make it easier to remove unneeded fonts from within the Office 2016 apps, but you can also clear all non-existent fonts from its lists! Microsoft didn't tell anyone about this, so it was a discovery process. I noticed in the 15.17 updates that it added a lot of .plist files to the application packages. They all have to do with the fixed font lists you see in Office 2016.

Close all Office 2016 apps if they're running. For each application package, right click on an app and choose **Show Package Contents**. In each app, drill down to **Contents/Resources**. It's easiest to do this next part in List view, so press Command+2 if you're not already in List view. Type "font". The list will jump to the first entry by that name, which is **fontFamilyImages.plist**. Highlight that item, scroll all the way down to **fontsInfo.plist** and Shift+click on that file. Then scroll up and Command+click on the folder **Fonts** to deselect it. That will leave 60 .plist files selected, all with *font* as the first part of the file name. Delete all of them.

The number of items to remove has changed slightly. In later updates to Office 2016, from **fontFamilyImages.plist** through **fontsInfo.plist**, the total is 56. Make sure to remember to deselect the **Fonts** folder within that group.

When you launch each Office 2016 app, they will open a bit more slowly the first time after doing this as it rebuilds its font list and caches. But now each app will only show fonts that actually exist on the drive, along with the few in each Office 2016 application package (assuming you've also removed each embedded DFonts folder).

15.22 through current

The Office 2016 15.22 update has changed things again. One better, the other worse. 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 have 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 noted above as before causes the Office apps to show no fonts at all in its lists, other than Cambria.

The steps for 15.22 through current are longer and more involved, but it can still be done. Since these instructions cover multiple minor point releases, just assume in this particular section that they apply to any release of Office 2016 that is version 15.22 or later. If Microsoft radically changes things again, I'll start a new section.

As above under the 15.17 notes, right click on an Office 2016 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 **fontImages2052.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.

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.

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 [TextWrangler](#). 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.

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

What to do with Suitcase

Suitcase X and X1:

Once you have your fonts where you want them and you are running Suitcase X or X1, you will need to reset the application. When any fonts are added manually, or by another application to an active Fonts folder, those fonts are automatically added as fixed entries in Suitcase X and X1. You cannot remove them within the application. Also, removing any of those fonts manually from the Fonts folder(s) they are in, as we are doing here, does not clear those entries. Suitcase will forever list all fonts which are removed manually as though nothing has changed. In addition, you can't remove the non-existent fonts from its list.

To fix Suitcase, go to the `~/Library/Preferences/` folder and remove the Suitcase preferences, which is a folder named "Suitcase" within the Preferences folder. Restart your Mac. Suitcase will be as it was when you first installed it, but now only fonts that truly exist will be listed. You will need to reset any other application preferences that are not the defaults. You will also lose all font sets you have created. Since my experience is almost exclusively with Suitcase, I don't know how manually removing fonts may affect other font managers.

None of this is necessary with the original Suitcase Fusion through Suitcase Fusion 7. Simply restart your Mac and Suitcase will correctly reflect the active fonts on your system. That said, occasionally Fusion will *not* fix itself after a restart. See the next following categories to repair the version of Fusion you are using.

Suitcase Fusion:

To manually reset Suitcase Fusion, go to the `~/Library/Application Support/Extensis/Suitcase/` folder, remove the file `Suitcase Font Database.suitcasevault` and restart your Mac.

Suitcase Fusion 2 through 7:

To manually reset Suitcase Fusion 2 through 7, go to the `~/Library/Extensis/Suitcase Fusion/` folder, remove the file `Suitcase Fusion.fontvault` and restart your Mac. You may need to enter the System Preferences and reselect the newly recreated database.

You can also replace this file through the System Preferences. Open the System Preferences and click on the Suitcase Fusion Core icon.

Click the Stop button to halt Suitcase. Once it stops, the buttons at the lower right will be active. Click "New..." and Suitcase will show the current file being used. Click "Save" to use the same name. Suitcase will warn you that you're about to replace the current database. Click "Replace". Restart your Mac. You may need to reenter the System Preferences to turn the Fusion Core on. You will only need to do this once after replacing the database. With version 4 or later, the Fusion Core is in Suitcase's preferences under the Type Core tab, rather than in the System Preferences.

Preserving fonts in the vault for Suitcase Fusion and Suitcase Fusion 2 through 7:

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. This would be especially bad if you also have the preference set to delete the original fonts (original version of Fusion only) after adding them to the vault and you don't have access to the originals. Losing your original fonts does not apply to Suitcase Fusion 2 through 7 as they do not have the option to delete fonts you add to the vault, but you will still lose all fonts stored in the vault if you remove the database. 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) In the Suitcase Fusion interface, change the pull down menu between the two left panes to "Suitcase Fusion Fonts". These are the fonts stored in the vault. Highlight all sets in the top left window. For Suitcase Fusion 2 through 7, your sets will be in the left pane under the "Font Library" heading. Highlight the first set and then Shift+click on the last set to select all. With any Fusion version, now 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 Collect button in Fusion, or the Choose button in Fusion 2 through 7. 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. Click Quit on Fusion's warning about shutting it down (Fusion 2 through 7 work differently and will simply close).
- 3) Go to your user account and remove the vault database noted above for Fusion, or Fusion versions 2 through 7. Relaunch Suitcase Fusion, or restart your Mac for Fusion 2 through 7.
- 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 and Courier 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. 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 Panther and Tiger; Courier and Helvetica must be present in some form. They do not however, have to be the .dfont versions supplied with OS X. Any form of Courier or Helvetica will do to satisfy OS X's needs, whether it is the supplied .dfont, Type 1 PostScript, Mac legacy TrueType, or OpenType version.

For Leopard through Yosemite, the .dfont, .ttf and .ttc supplied versions of Helvetica and Helvetica Neue are required for various apps supplied with OS X. There is also Helvetica Light as a separate font in Snow Leopard (it is folded into Helvetica.dfont in Lion through Yosemite). You can still use the following instructions to remove Apple's .dfont, .ttf or .ttc fonts from the System folder, but you should

first place copies of the OS X supplied fonts in another location. That way, you can use your font manager to activate your PostScript fonts and deactivate the .dfont/.ttc/.ttf fonts when needed. Then deactivate the PostScript fonts and activate Apple's fonts for normal use of your Mac. For El Capitan, Helvetica is only used for Notes, but is critical 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 6 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 and Courier in Panther and Tiger.

Remove Courier.dfont and Helvetica.dfont from the `/System/Library/Fonts/` folder. You will need Administrative access to remove these two fonts. Then permanently activate Type 1 PostScript or PostScript OpenType versions of Courier and Helvetica with your font manager, or place copies of your preferred fonts directly in the `/Library/Fonts/` folder.

Despite its technically different name, earlier OS X versions of Courier New conflicted with the PostScript Courier. Remove the .dfont version of Courier New also and use a standard PostScript Type 1 or PostScript OpenType copy of Courier instead. In Tiger and Leopard, the internal names of the OS X supplied versions of Courier New have been fixed so as not to conflict with Courier. So both versions of Courier can be on your system if you wish.

You can also place permanent copies of Courier and Helvetica in any of your hard drive folders listed in the next section. Unlike OS 9 or earlier, you can place folders of fonts into a Fonts folder. OS X will see and open all fonts in any subfolders. The disadvantage of opening fonts this way is that they will not be available to programs running in Classic. Use a font manager such as Suitcase (versions X or X1), MasterJuggler (PowerPC only) or FontAgent to overcome this limitation.

Removing Helvetica, Helvetica Neue and Helvetica Light in Leopard through Mountain Lion.

Apple has gone to fairly great lengths in Leopard through Mountain Lion to save the user from themselves by protecting certain fonts. While this is a wonderful safety net for the typical user, it's a headache for professional printers and prepress shops who *must* use PostScript versions of Helvetica. Especially since Font Book always favors the Apple versions when using Resolve Conflicts. It doesn't even give the option of choosing one over the other. On top of that, if you try to open a Type 1 PostScript font, only these styles not already active in Apple's versions will appear. I activated a Type 1 PostScript version of Helvetica through Suitcase Fusion (before the Fusion 2 through 7 versions) with Apple's version still in place. Even with the option to allow Suitcase to manage system fonts on and Font Book's option to "Alert me if system fonts change" off, Suitcase still could not override the Apple supplied versions, leaving only one PostScript typeface active that didn't already exist in the .dfont copy.

Because Leopard through Mountain Lion keep resurrecting them, removing Helvetica.dfont and HelveticaNeue.dfont (HelveticaNeue.ttc and Helvetica Light in Snow Leopard) is not a simple matter, but it can be done. To do so, follow the steps below.

1) Go to the `/System/Library/Fonts/` folder. Copy Helvetica.dfont, HelveticaNeue.dfont / (HelveticaNeue.ttc and Helvetica Light - Snow Leopard only) to another location for normal use of your Mac when you do not need a Type 1 PostScript version active. Helvetica.dfont, HelveticaNeue.dfont / HelveticaNeue.ttc and Helvetica Light do not need to specifically be in the `/System/Library/Fonts/` folder in order to work correctly for Leopard through Mavericks. Therefore, you can activate them with your font manager in the usual fashion.

2) Go to the `/System/Library/Frameworks/ApplicationServices.framework/Versions/A/Frameworks/ATS.framework/Versions/A/Resources/ProtectedFonts/` folder. Delete the files Helvetica.dfont, HelveticaNeue.dfont (HelveticaNeue.ttc and HelveticaLight.ttf in Snow Leopard). **Do not remove any other fonts!** Restart your Mac.

3) If you use Font Book, open Font Book and highlight both Helvetica, Helvetica Neue and Helvetica Light (Snow Leopard). Disable the fonts using the menu or by pressing Command+Shift+D. With both files still highlighted, right click (or Ctrl+click with a one button mouse) and choose "Remove Fonts". The fonts should disappear from Font Book's list and the fonts themselves from the `/System/Library/Fonts/` folder. In Lion and later, Font Book will not remove any fonts in the System folder, even though the choice to Remove Fonts is still an option. If the fonts remain in the `/System/Library/Fonts/` folder, you will have to manually remove them. You will need your Administrator password to do so. Restart your Mac.

4) If you do not use Font Book, go directly to the `/System/Library/Fonts/` folder and delete Helvetica and Helvetica Neue (also Helvetica Light in Snow Leopard). Enter your Administrator password when prompted to complete the action. In Lion or later, do not remove HelveticaNeueDeskUI.ttc.

5) Restart your Mac. This is important as the OS does not release fonts removed from the System folder. They remain active even in the

Trash. You must restart in order to release them, otherwise you will still be unable to use your third party Helvetica fonts.

With the backup fonts for Helvetica.dfont, HelveticaNeue.dfont / (HelveticaNeue.ttc and Helvetica Light in Snow Leopard) removed from the ProtectedFonts folder, they will no longer keep reappearing in the [/System/Library/Fonts/](#) folder. You will now be able to activate full copies of your preferred Helvetica fonts.

Courier is no longer a required font for OS X in Leopard, 10.5 through Mavericks, 10.9. As long as you have Administrative privileges, you can remove the supplied .dfont copy of Courier from the [/System/Library/Fonts/](#) folder.

Removing Helvetica fonts in Mavericks through Sierra.

The ProtectedFonts folder no longer exists in Mavericks through Sierra. Copy Helvetica.dfont and HelveticaNeue.dfont to another location and then remove them from the [/System/Library/Fonts/](#) folder. See the bottom of this section regarding HelveticaNeue and Office 2016.

Suitcase Fusion 2 through 7 and Helvetica fonts.

These versions of Suitcase can automatically circumvent the Apple Helvetica fonts.

- 1) Turn off the check box in Font Book's preferences to warn when system fonts have changed (Mountain Lion and earlier). Close the preferences. 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 2 through 7'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.

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 2 through 7 automatically deactivate 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.

A bug in El Capitan prevents legacy Mac TrueType and Type 1 PostScript fonts from working in Office 2016. Extensis tracked it down and has posted workarounds for the [client version](#) of Suitcase Fusion 7, and a similar article for the [Universal Type Server](#).

One sure workaround in Suitcase Fusion 7 is to use the preference, *Copy added fonts to the Vault*. Type 1 PostScript and legacy Mac TrueType fonts will then work correctly every time.

This bug has been fixed as of 10.11.3. I was able to activate older Mac legacy TrueType fonts and Type 1 PostScript fonts in Suitcase using the "in place" option, and by dragging and dropping the fonts into the [/Library/Fonts/](#) folder.

General information for third party programs.

Microsoft has made this a bit difficult with the introduction of Office 2016, but further testing has revealed two easy workarounds. There's no need to move the macOS or OS X supplied version of HelveticaNeue.dfont around or remove it from the [/System/Library/Fonts/](#) folder. Office 2016 continues to function if you turn the Type 1 PostScript versions on and off even while the Office 2016 apps are running.

- 1) Add your Type 1 PostScript versions of Helvetica and Helvetica Neue as separate sets in your font manager. Turning them on and off correctly activates the T1 PS versions over the macOS or OS X supplied versions. How well this works depends on what font manager you're using.
- 2) You can manually drag and drop the Type 1 PostScript sets of Helvetica and Helvetica Neue in and out of the Fonts folder of your user account. The fonts you place in this folder will override the system fonts until they are removed.

In either case, you'll notice that the fonts displayed in Office 2016 change their shape slightly as you turn the Type 1 PostScript versions on or off and Office 2016 flips over to the active version of Helvetica Neue, but the Office 2016 apps don't crash. This may also partly be a fix

with the 15.15 updates to Office 2016. So make sure you have it up to date.

The Adobe apps aren't fussy at all. 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 and OS X use 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 manger, all like named versions must be removed from the higher priority folders. (Suitcase Fusion 2 and later can override the system fonts).

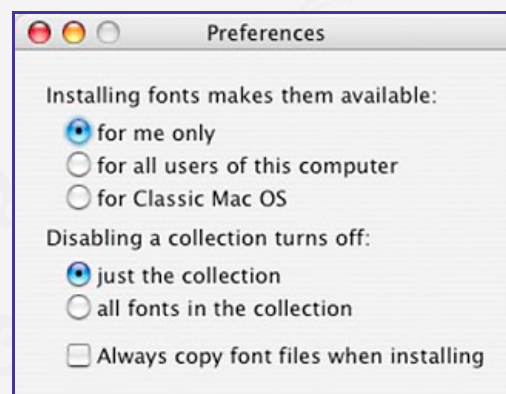
OS X's Font Book has preferential treatment here. By that, I mean fonts you activate using Font Book will get a higher priority over other font managers as it uses the folders listed above to activate and deactivate fonts.

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. The advantage of Suitcase (versions X and X1) and Master Juggler is that fonts are activated from wherever they are located on the hard drive, eliminating the possibility of a font being corrupted during the copy method Font Book uses (prior to 10.5). These three particular third party font managers also have the advantage of activating fonts for OS X and the OS 9 Classic environment simultaneously, with the exception of .dfonts as noted in the last paragraph of section 8.

Suitcase Fusion 2 through 7 can also activate fonts in place, but no long support activating fonts for Classic. FontAgent and MasterJuggler are the only remaining font managers which can simultaneously open fonts for OS X and Classic.

Font Book in Panther, 10.3 (before 10.3.5)

If you use Font Book to manage your fonts, it copies the font(s) selected into one of the folder locations listed above depending on which radio button you choose in the application. When you remove the font or fonts, Font Book deletes the fonts from the location they were copied to.



Font Book preferences in Panther, pre 10.3.5

If you choose **for me only**, the font(s) are copied to the ~/Library/Fonts/ folder, automatically giving them the highest priority over any same named font in the hierarchy. If you switch users, the font(s) will not be active under other accounts on that Mac since they are in your user folder.

If you choose **for all users of this computer**, then the font(s) are copied to the `/Library/Fonts/` folder. In this case, priority drops down one rung in the hierarchy and will be active when switching to another user, even though they didn't open the font(s) since they will reside in the common `/Library/Fonts/` folder.

Lastly, choosing **for Classic Mac OS** will copy the font(s) into the OS 9 `/System Folder/Fonts/` folder. Not only will fonts activated this way be available to applications running in the OS 9 Classic mode, but also to OS X native programs. When you remove fonts opened with this last choice, the fonts remain available to Classic applications until the Classic environment is restarted even though they are removed from the `/System Folder/Fonts/` folder. Fonts opened with Suitcase (versions X or X1), FontAgent, or MasterJuggler are also made available to applications running in Classic and OS X simultaneously without copying.

Font Book in Panther 10.3.5 through Tiger, 10.4

Starting with OS X Panther 10.3.5, Font Book changed the way it handles closed fonts. Previously, it would delete the fonts from the active font folder when you disabled them. Its new method is to leave the fonts where they are and use a separate system function to denote that they are supposed to be closed. This causes problems with most third party applications in versions of OS X before 10.4.3. Since the fonts are left in a Fonts folder, the underlying Unix system still considers them active. So "disabled" fonts still appear in almost all non Apple applications such as those by Adobe, Microsoft, Quark and others. There is no workaround in OS X before 10.4.3 other than to discontinue use of Font Book and use a third party font manager such as those listed in section one.

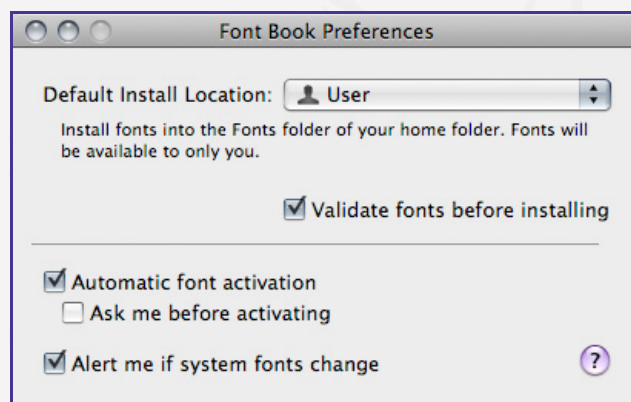
There are only two preferences in this version of Font Book. Where you want activated fonts copied to and whether or not to validate them (check for corrupt fonts). The only locations available when activating fonts are the Fonts folder of your user account at `~/Library/Fonts/`, or the global `/Library/Fonts/` folder. Any new fonts you activate will be copied to whichever Fonts folder you chose in Font Book's preferences if they aren't already there. When you deactivate the fonts, they remain in the folder they were copied to.

OS X 10.4.3 and later resolved the problem of having fonts in the `/Library/Fonts/` or `~/Library/Fonts/` folders. All applications I tested with did indeed respond to the fonts being deactivated with Font Book. Even older applications that were written before Font Book existed. This change appears to be at the system level since Suitcase Fusion and FontExplorer X Pro can successfully do the same thing.

A few readers note that they are still having issues with fonts remaining on after being marked as disabled in Font Book. But I think that's a matter of a corrupt database. It's very common for fonts to remain active, or unable to be activated if Font Book's database is damaged. See just below for information on how to reset Font Book.

Font Book in Leopard, 10.5, Snow Leopard, 10.6 and Lion, 10.7

Font Book adds a few more choices to Font Book in Leopard and Snow Leopard, but it remains basically the same application with the same methods. Newly activated fonts are copied to one of the two available Font folder locations. New options include (1) the option to validate fonts when activated (check for errors) and (2) to automatically activate fonts a document needs that you have opened, if possible. There is a sub choice to 'ask first' before activating in such an instance. It also has an option to alert you if the system fonts have changed, which is on by default.



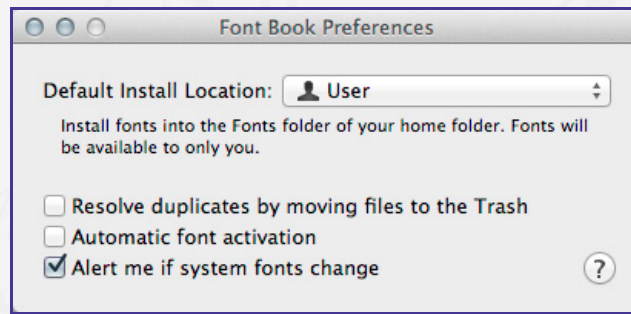
Font Book preferences in Leopard

Starting with Leopard, Font Book includes the ability to open fonts in place. To do so, you need to choose **File > New Library** from the menu. Make sure the Library you created is highlighted and add your fonts. Rather than copying every font you activate to the Fonts folder chosen in Font Book's preferences, the fonts are activated from wherever they reside.

Lion adds a font conflict resolution screen. See the font reviews in section 16 for a more detailed explanation of this feature.

Font Book in Mountain Lion, 10.8

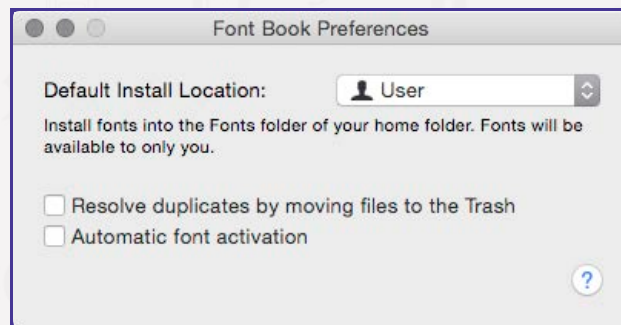
The preferences add a new option and removes two others. Instead of having Font Book simply disabling conflicting fonts, you can choose to have them sent to the trash to permanently resolve the conflict. This is not without problems, though. See section 16. The removed options are (1) validating fonts before installing and (2) asking before automatically activating fonts. The latter option is now simply on, or off. Font validation hasn't been removed, it's now an automatic process which happens any time you add new fonts.



Font Book preferences in Mountain Lion

Font Book in Mavericks, 10.9 through El Capitan, 10.11

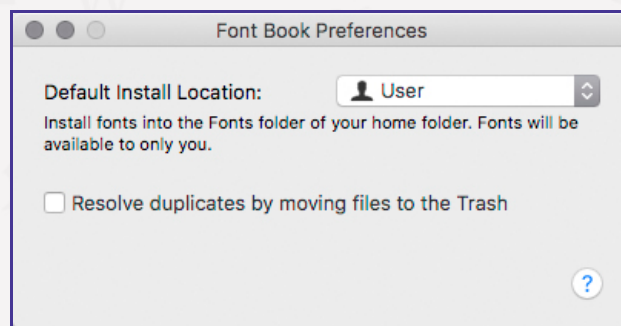
The preferences have been simplified even further. There is no longer an alert for system font changes.



Font Book preferences in Mavericks through El Capitan

Font Book in Sierra, 10.12

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

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.

Panther through Sierra each have their own methods for resetting Font Book. Dr. Smoke, (as he is known at the Apple Discussions forums), is the noted author of *Troubleshooting Mac OS X*. He has kindly made the information for resetting Font Book freely available at his web site. See [Undoing Font Book](#) for detailed instructions for the version of OS X you are using.

There are no instructions for Lion through Sierra at Dr. Smoke's site, but the Snow Leopard instructions listed there apply to them also. The instructions remove the font cache files themselves for the current logged in user, but does not reset Font Book's database.

To reset Font Book completely in Snow Leopard, 10.6.x through Sierra, 10.12.x, do one of the following.

1a) 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 or OS X 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. This method works for all versions of the Mac OS.

1b) For a while, Font Book's database file, [com.apple.FontRegistry](#), was buried deeply in the `/private/var/folders/.../-Caches-/` folder. In later versions of OS X, and in macOS, it's much easier to remove. 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](#)

In El Capitan and Sierra, [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 *may* also apply to Yosemite. This method of resetting Font Book is faster than 1a, where you have to restart twice.

In Leopard or earlier, any font sets you have created will be gone. Also, all fonts in the three main Fonts folders (System, Library, your user account) will now be active, regardless of their state beforehand.

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. The next time you launch Font Book, it will create a new database.

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 and OS X will read this database (if it exists) during startup, or when logging into another account. Based on what that database says, macOS or OS X 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.

1) Open Font Book, then its preferences. Uncheck the box for "Alert me if system fonts change" (this option does not exist in Mavericks through Sierra). Close the preferences and shut down Font Book. In Snow Leopard or older, put the Font Book application in the trash and delete it. See below on the various methods of removing Font Book for later versions of the Mac OS.

2) Follow steps 1a or 1b above. The main goal in this step is to remove the orphaned Font Book database. With the Font Book application no longer on the hard drive, a new database cannot be created. Which is what we want.

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

Removing Font Book in Lion through Yosemite

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 Mac OS X (macOS)*. Don't take this message to mean Font Book is now an immutable part of the OS. Lion through Sierra 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) Open Font Book, then its preferences. Uncheck the box for "Alert me if system fonts change" (this check box is not in Mavericks through

Sierra). Close the preferences and shut down Font Book.

2) 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 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.

Then follow the steps 2 and 3 above under **Removing Font Book**.

Removing Font Book in El Capitan and Sierra

The new System Integrity Protection makes this more of a nuisance, but still isn't all that difficult to get around. Refer to the two methods of removing fonts outlined in El Capitan of section one and use either procedure to remove Font Book.

Then follow the steps 2 and 3 above under **Removing Font Book**.

Required fonts for OS 9 (Classic)

For the OS 9 Classic mode, look in the `/System Folder/Fonts/` folder. The only ones that must be present are:

- Charcoal
- Chicago
- Geneva
- Monaco

If your Mac is capable of booting directly into OS 9, you can use Suitcase (version X or earlier), Adobe Type Manager Deluxe or Master Juggler to open any other fonts you prefer to have open (Font Book is a OS X application and will not run directly under OS 9). Otherwise, these four fonts are the only ones that must be present for OS 9; whether booted into OS 9, or running as Classic within OS X. If you have any other fonts in that folder, move them out. Particularly Helvetica and Courier, which with a default install of OS 9 will be in that Fonts folder and will conflict with the OS X or PostScript versions you may wish to use instead.

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, Master Juggler or other font manager. You will also find a subfolder at `/Library/Application Support/Adobe/Fonts/Requd/`. 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 OS X, 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.

You cannot use .dfonts in the Classic environment. Legacy OS 9 TrueType fonts have their data stored in the resource fork of the file, while a .dfont stores the data in the data fork. Hence the name .dfont, short for data fork font. If you open a .dfont with any font manager, your OS X applications will be able to use them, but OS 9 Classic applications will not understand how to read or use these fonts, even if they have been copied to the OS 9 `/System Folder/Fonts/` folder. If there is any particular .dfont you wish to use in Classic, you can convert it to an OS 9 TrueType suitcase font using a utility such as [dfontifier](#). This application quickly and easily creates a new font that OS 9 can use by dragging and dropping the .dfont you want to use in Classic onto the dfontifier application. You then open the converted font for Classic. The program dfontifier is no longer being developed by the author and will work only in OS X through Tiger 10.4.x. For other means of translating fonts, you can use [FontLab Studio](#), [Fontographer](#), [TransType](#) or the free open source [FontForge](#).

The retail Snow Leopard disk includes fonts intended for iWork, even if you didn't purchase the Box Set. In Snow Leopard, where they first appeared, I had originally thought this may have been a production error; but these same fonts are installed by Lion through Sierra. So for whatever reason, 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 25 .dfonts (24 in Lion, 25 in Mountain Lion, 28 in Mavericks and Yosemite), which comprise a total of 37 individual type faces (33 in Lion, 37 in Mountain Lion, 53 in Mavericks, 107 in Yosemite counting its extra subfolders). 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. This isn't a bonus for everyone though. These fonts will only activate in Snow Leopard or later. Any earlier versions of OS X cannot use them. If you wish to obtain these fonts for Leopard or earlier, you can install the iWork '09 trial (may no longer be available). It will install copies of the same 25 .dfonts found in Snow Leopard, which do work in previous versions of OS X.

If you really feel like digging, Lion also installs 23 .ttf fonts intended for X11. They are in the `/user/X11/share/fonts/TTF/` folder. These fonts are not installed in Mountain Lion, Mavericks or Yosemite.

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.

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 6 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 are disadvantages to this method though. One is that fonts activated this way will not be available to programs running in Classic. In addition, 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
AGarBolIta
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.

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.

While OS X, 10.5.x Leopard no longer supports the Classic environment, it will still see any fonts in an OS 9 [/System Folder/Fonts/](#) folder. So if you are experiencing font conflicts and can find no cause for them in the standard macOS or OS X Fonts folders, check to see if an OS 9 folder for Classic is still on your system. I can no longer test it, but this also may apply to Snow Leopard and later.

Conflicting internal names is exactly what the font problem is between Apple-supplied versions of Helvetica in Leopard through Sierra 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 5.

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 recently 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 and OS X

Fonts which can be used in macOS and OS X:

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

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 2 through 7 also note 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**.

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 or OS X:

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 and OS X:

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.

Fonts for Classic (OS 9):

Because OS 9 expects fonts to be created a certain way, it cannot use some fonts that macOS and OS X can. Legacy TrueType from OS 9 and earlier and Type 1 PostScript fonts have their data in the resource fork, which is where OS 9 expects to find the font data. The Mac .dfont, while it is a TrueType font, cannot be used in OS 9. This is due, at least in part, because the data is in the data fork of the font; hence OS 9 sees nothing. The same is true of Windows .ttf TrueType fonts. The data in is the data fork rather than the resource fork, so OS 9 again sees nothing. Very few applications for OS 9 were ever updated to understand OpenType fonts. Windows Type 1 PostScript fonts are completely incompatible and cannot be used by either OS 9, OS X or macOS, with the above exception for the Adobe applications.

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 10.12.x Sierra, FontExplorer X Pro 6, MasterJuggler 3.0.4, Suitcase Fusion 7 (18.2.1), RightFont 3.2.3, and TypeDNA 2.7.1. Since Classic (OS 9) went away with Leopard in late 2007, I'm no longer going to mention it.

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 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.

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.

Master Juggler

Pros:

- 1) Can create full sets that include font names already in other sets.
- 2) Activates fonts in place. No need to create copies of the fonts you activate.

Cons:

- 1) Makes no attempt to resolve font conflicts. Nor does it give you a function to do so. You must search through the list of fonts in its interface manually to find and resolve them yourself.
- 2) Makes no attempt to stop you from activating fonts that will conflict.
- 3) No auto activation feature.
- 4) Cannot deactivate fonts in the `/Library/Fonts/` or `~/Library/Fonts/` folders.
- 5) Is still a PowerPC only app. Cannot be used in OS X Lion, 10.7.x or later.

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.

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 and OS X, so you can play around with it as much as you want. FontExplorer X Pro, Suitcase Fusion 7, 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. MasterJuggler 3 is the only offering that requires you to purchase the product in order to see what it can do. No trial software is available. MasterJuggler also hasn't received an update in years.

Suitcase Fusion 2 through 7 have 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. There is no longer an option in the preferences as to what to do with the system fonts, but if you activate (for example) a Type 1 PostScript version of Helvetica (which will conflict with Apple's versions) Suitcase Fusion 2 through 7 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 5 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 7, 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 Leopard through Sierra 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.

If opening fonts for OS 9 (Classic) is still important for you, and you don't like manually moving fonts in and out of OS 9's `/System Folder/Fonts/` folder to activate/deactivate your fonts, then your only choices for font managers are FontAgent 6 and MasterJuggler 3. With the Intel based Macs making OS 9 virtually obsolete, I wouldn't be surprised if a future release of these two titles removes that ability. Starting with OS X Leopard, Apple has eliminated support for OS 9 entirely; so whether or not a font manager still supports Classic is a moot point if you're using Leopard, 10.5 or later.

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 little, or drastically different from older versions.**

1) Suitcase Fusion 7: 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 7 is incredibly stable. It is almost impossible to make it crash, even with the worst fonts. Suitcase Fusion 2 and later have eliminated the potentially dangerous option of deleting your original fonts after adding them to the vault. Adding fonts to the vault now always leaves the originals intact.

Handling of conflicting fonts has been simplified. 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. This is actually a fairly recent change in this program's behavior. Before Fusion 2, you used to be able to choose font-by-font whether to activate or skip on each conflicting font. Perhaps the function will return if enough users request it. Fusion 3 added a font cache cleaning option within the interface. Version 4 introduced Quickmatch. To see what other goodies Extensis has added to Fusion 7, download the user guide from their web site.

When Extensis acquired Diamond Soft, 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. Suitcase Fusion 7 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 7.

Suitcase Fusion 7 continues to be my favorite font manager. The interface and preferences are even more intuitive than the original OS X versions or the initial version of Fusion. If you've used Suitcase for a long time, you feel like something's been taken away in the sparse looking interface of Fusion 2 through 7, but it's still a very powerful program with even better options and features; while at the same time removing other interface choices that really weren't needed. The new interface makes it even easier and almost goof proof to use while still being, I think, the best font manager available for pros and novice users alike.

2) FontExplorer X Pro: If Suitcase no longer existed, I wouldn't hesitate a moment to use this manager. 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. I'm sure it's just a minor bug, but 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 show you that fonts are disabled that you know for a fact are 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 7 and FontAgent have a method of avoiding this possibly very expensive error. Regardless of that, FontExplorer X Pro has a large number of options to fully control your fonts.

3) 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 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 the point where 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.

Version four'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.

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.

Lion through Yosemite adds a greatly appreciated font conflict resolution interface. Instead of just blindly turning one font or the other off (you have no idea which one, and it doesn't tell you), it presents a screen of the conflicting fonts. You can choose each in turn and select "Reveal in Finder". The folder the font is in will open and the file itself is highlighted. You can now make an informed decision as to which font should be deactivated.

Now if only it worked. The conflict resolution screen isn't as straight forward as you would think. Just like disabling a conflicting font from the main screen, the currently active font is *always* the one to remain active. Even when using the option to **Resolve Manually**. It doesn't matter which font you highlight, the inactive font remains that way. So manual doesn't do anything different than automatic. **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. This dangerous function remains unchanged in Yosemite.

Font Book's conflict resolution screen in El Capitan and Sierra 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 one 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 older versions of Font Book's interface, which aren't anywhere in the various Fonts folders, used to be buried in the Font Book application package itself. Stub fonts have returned in Sierra 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 and Sierra. 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.

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. However, Font Book is slowly getting better.

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 Aaaaaaa. 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 use in a production environment.

7) Master Juggler: I rather expected a font manager from a company that produces the indispensable DiskWarrior to be much better than this. A font manager that won't stop you from activating conflicting fonts even from its own interface is useless. You may as well not use a font manager at all and manually move fonts in and out of the `/Library/Fonts/` or `~/Library/Fonts/` folders.

In its favor, MasterJuggler is a very stable program that does what it does do very well with full font sets and handling even bad fonts well. But it is very much a font manager that requires the user to fully understand how font conflicts occur, how to avoid them and how to resolve them on your own.

It seems Alsoft has abandoned MasterJuggler even though they still sell it. Its last update was 3.0.4 on June 1, 2005 and is still a PowerPC only app, which makes it completely unusable in Lion, 10.7.x or later. Suitable only for users of Snow Leopard, 10.6.x or older.

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.

For users of Snow Leopard, 10.6.x or older, there's also [Font Finagler](#). Download and run the application. First click the "Inspect Font Cache Files" button, then the "Clean Font Cache Files" button. You must then restart your Mac.

Suitcase and FontExplorer X Pro both come with cache cleaning tools. With Suitcase Fusion through Fusion 2, it's the bundled Font Doctor. The option in Font Doctor is under **Tools > Clean Font Caches**. Suitcase Fusion 3 through 7 have included a cache cleaning choice in the main application so you don't have to launch Font Doctor. With Suitcase Fusion 3 through 7 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:

One fairly common problem that occurs with corrupt font caches (particularly in Tiger for some reason) are fonts in Dashboard widgets suddenly being drawn as outlines instead of solid text. Clearing the font cache files usually fixes it.

There is a right way and a wrong way to remove cache files. Font Finagler and 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 7, 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 X and 2004, the location is:

`~/Library/Preferences/Microsoft/Office Font Cache (11)`

For Office 2008, the location is:

`~/Library/Preferences/Microsoft/Office 2008/Office Font Cache (12)`

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 10.5.x Leopard through 10.12.x Sierra, 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 and OS X. 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 Lucida Grande while cleaning out my fonts. Now my Mac won't boot. Help!" (substitute Lucida Grande with HelveticaNeueDeskInterface for Yosemite, and the San Francisco set for El Capitan or Sierra).

A: There are ways to get this critical font back onto your Mac.

1) Turn your Mac on and hold down the mouse button. This will cause the DVD/CD drive door to open so you can insert the OS Install disk that came with your Mac, or a retail copy if you purchased it separately. After placing the disk in the tray, close the drive by nudging the extended tray with your finger. Press and hold the power button until the Mac turns off. Wait a few seconds and press the power button again. Immediately hold down the C key to boot to the Install disk in the DVD/CD drive. Do not release the key until it is clear you have started up your Mac to the disk. Run the OS installation, making sure to choose the Archive and Install option. Also make sure to choose the options to maintain your user settings. With Snow Leopard, run the standard install to preserve your third party apps and settings. Once the install has finished, you can restart and your Mac should once again be able to boot to the hard drive. Your system will be at whatever level the disk you have is. So if your Install DVD/CD disk is 10.3.3, that's what version of OS will be on your Mac. You will need to download the appropriate Combo Updater from Apple's site to restore the system to a higher version.

2) This method is much less invasive or time consuming. The only drawback is that you need access to another Mac running the same basic version of OS X (Panther, Leopard, etc.), and your Mac must support FireWire Target Disk Mode. Macs which support this function are listed in the Apple Knowledge Base Article [HT1661](#). Place the non functioning Mac next to a working Mac and connect the two directly to each other using a standard FireWire 400 or 800 cable. It doesn't matter if the Macs are on or not. FireWire is a hot plug-able connection and so 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 FireWire Target Disk Mode. Keep holding the T key until a large FireWire 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 Lucida Grande font 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 FireWire 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 FireWire cable. When you start your repaired Mac again, it should boot. Once started, run Disk Utility to Repair Permissions as the fonts in the System folder will have incorrect settings.

3) The third method is similar to the second. And that is to have an external FireWire drive with the same version of macOS or OS X installed. To use this method, your Mac must support booting to a FireWire device. Macs that support this feature are listed in the Apple Knowledge Base Article [HT2699](#). Intel based Macs also support USB booting, as per article [HT1948](#).

Have your non functioning Mac off. Press and hold the power button if necessary to power the computer down. Connect the external FireWire (can be FireWire 400 or 800) or USB 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 LucidaGrande.dfont from the external drive to your main drive as described in method two. Copy HelveticaNeueDeskInterface.ttc to the main drive for Yosemite. 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.

4) 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 the letter A 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 cause is that your font cache data is damaged. When that happens, the OS opts for the last resort to display something, *anything* there. Appropriately, this text is from the system font LastResort. Usually, 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 that still doesn't help, you may need to reinstall the OS.

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.

First, download [Pacifist](#). Insert the first OS installation CD or DVD. Start the utility and click on the "Open Mac OS X Install Packages" button. Keep feeding the application the CDs or DVDs of your Panther through Snow Leopard Install disks as it asks for them. If you have a set of DVDs that came with your Mac, the first disk should be the only one you need. After Pacifist has finished reading all of the packages, select the package that has the fonts you want installed 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 OS X to get your fonts back where they belong.

Notes on restoring fonts on Macs that shipped with (or have been upgraded) to Lion through Sierra:

These Macs do not come with any 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 Lion or later is at least 4 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/`. 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.

So whenever you get the opportunity to download an entire Lion, Mountain Lion, Mavericks, Yosemite, El Capitan or Sierra system 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.

It's quite a bit different as Apple does not sell a CD or DVD for Lion through Sierra. Lion was available on a flash drive. The following instructions assume you copied your Lion through Sierra download to an external drive or other location to save for future use. I haven't seen a flash drive version of Lion, so I can't comment on how that may work.

Font locations for Lion, Mountain Lion, Mavericks, Yosemite, El Capitan and Sierra:

Right click on the Lion, Mountain Lion, Mavericks, Yosemite, El Capitan or Sierra installer and choose **Show Package Contents**. Double click the **Contents** folder, and then the **SharedSupport** folder. Double click the file **InstallESD.dmg** to open the disk image. A disk image drive icon will appear on the desktop. Double click the desktop icon to open it. In the resulting window, double click the **Packages** folder. Open the **Essentials.pkg** with Pacifist. You can do this by dragging and dropping the .pkg file onto Pacifist's interface.

For the System folder fonts, expand **Contents of Essentials.pkg > System > Library > Fonts** and highlight the folder "Fonts" by clicking on it once. Click "Install" at the upper left of the Pacifist window.

For the Library folder fonts, expand **Contents of Essentials.pkg > Library > Fonts** and highlight the folder "Fonts" by clicking on it once. Click "Install" at the upper left of the Pacifist window.

Font locations for the Leopard and Snow Leopard retail disks:

For the System folder fonts, expand [Contents of OSInstall.mpkg > Contents of EssentialSystemSoftware > Contents of EssentialSystemSoftwareGroup > Contents of BaseSystem.pkg > System > Library > Fonts](#) and highlight the folder "Fonts" by clicking on it once. Click "Install" at the upper left of the Pacifist window.

For the Library folder fonts, expand [Contents of OSInstall.mpkg > Contents of EssentialSystemSoftware > Contents of EssentialSystemSoftwareGroup > Contents of Essentials.pkg > Library > Fonts](#) and highlight the folder "Fonts" by clicking on it once. Click "Install" at the upper left of the Pacifist window.

The Additional Fonts are in two separate locations. Expand [Contents of OSInstall.mpkg > Contents of AdditionalFonts.pkg > Library > Fonts](#), or [Contents of OSInstall.mpkg > Contents of AdditionalFonts.pkg > System > Library > Fonts](#), depending on the set of additional fonts you want to restore.

Font locations for the Tiger retail disks:

For the System folder fonts, expand [Contents of OSInstall.mpkg > Contents of BaseSystem.pkg > System > Library](#) and highlight the folder "Fonts" by clicking on it once. Click "Install" at the upper left of the Pacifist window.

For the Library folder fonts, expand [Contents of OSInstall.mpkg > Contents of Essentials.pkg > Library](#) and highlight the folder "Fonts" by clicking on it once. Click "Install" at the upper left of the Pacifist window.

Fonts locations for the bundled Tiger and retail Panther disks:

For the System folder fonts, expand [Contents of OSInstall.mpkg > Contents of EssentialSystemSoftware.mpkg > Contents of BaseSystem.pkg > System > Library](#) and highlight the folder "Fonts" by clicking on it once. Click "Install" at the upper left of the Pacifist window.

For the Library folder fonts, expand [Contents of OSInstall.mpkg > Contents of EssentialSystemSoftware.mpkg > Contents of Essentials.pkg > Library](#) and highlight the folder "Fonts" by clicking on it once. Click "Install" at the upper left of the Pacifist window.

You will also find a main package named [Contents of Fonts.mpkg](#). You do not need to restore these fonts unless you have need of the extended set of Asian and foreign language fonts. Also, you can expand each "Fonts" folder in Pacifist to select individual fonts if you only want to restore one, or a few.