



Installing Pro Tools 10 on macOS Sierra

Introduction

The Pro Tools 10 installer runs multiple checks before allowing you to install the program onto your system. One of them tests the version of your operating system and if it's above 10.8, an error message is shown. Some people got the nifty idea to change the OS-version saved in the macOS system files. Which is really a terrible idea and can potentially brick your computer. Don't do it.

So i've dug around a little and found a way to circumvent the install-check in a clean way. You'll need to use the terminal but it's really not that hard.

Disclaimer

You will do the following steps at your own risk. I am not to be held accountable for any damage you cause to your system. And if you are following this guide correctly, there is no way any damage can occur.

I do not endorse hacking applications but since the check that prevents installing Pro Tools 10 on newer systems is mostly arbitrary and the program costs a lot of money the steps described are reasonable.

Reading a guide from start to finish before executing the commands is always a good idea and also: even if you can now successfully install Pro Tools 10 on your system, it is still not supported by Avid and may crash regularly, have other bugs or not run at all.

Notice

The \$ sign in the listings below is not to be typed in. It's just a convention for marking text that belongs into a console application.

What follows is a short summary of this guide. Don't worry if you don't understand everything. All steps will be covered in detail afterwards.

Summary

Extract the installer .pkg file with 'pkgutil' and change the contents of the 'Distribution' file so that the variable 'resultOS' of the function 'installationChecks' is set to 'true'. Then repackage the extracted contents with 'pkgutil' into a new installer. Copy the other files from the install-volume into the same folder and run the new installer.

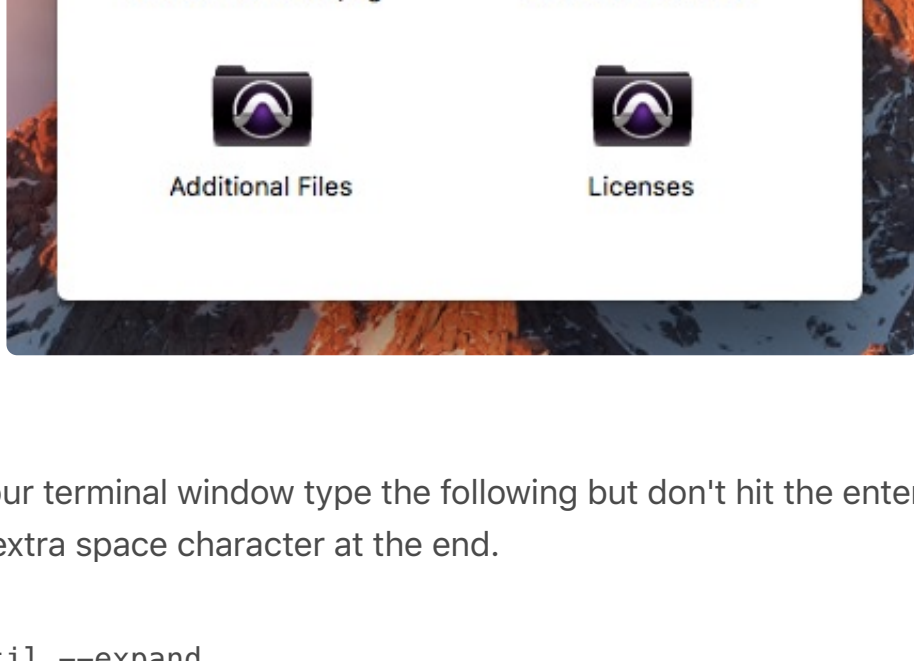
Extracting The Installer

To erase the version-check from the installer, we'll have to open it first. To do this apple provides a nice utility named 'pkgutil'. It can unpack a .pkg installer much like a zip file. Open the terminal of your Mac (it's in the Utility folder inside your Application folder) and let's get started.

First, go to your home folder and create a folder to save the extracted installer into.

```
$ cd ~/
$ mkdir Installer
$ cd Installer
```

Open the .dmg file or disk containing the installer files of Pro Tools 10. It contains the 'Install Pro Tools.pkg' file we need and also some other files needed during the installation. But we'll come back to them later.



Back in your terminal window type the following but don't hit the enter key. Also type one extra space character at the end.

```
$ pkgutil --expand
```

Next we need the path to the installer file. We can enter it by dragging and dropping the 'Install Pro Tools.pkg' file onto the terminal window. This should fill in the path and should now look something like this (still don't hit enter yet):

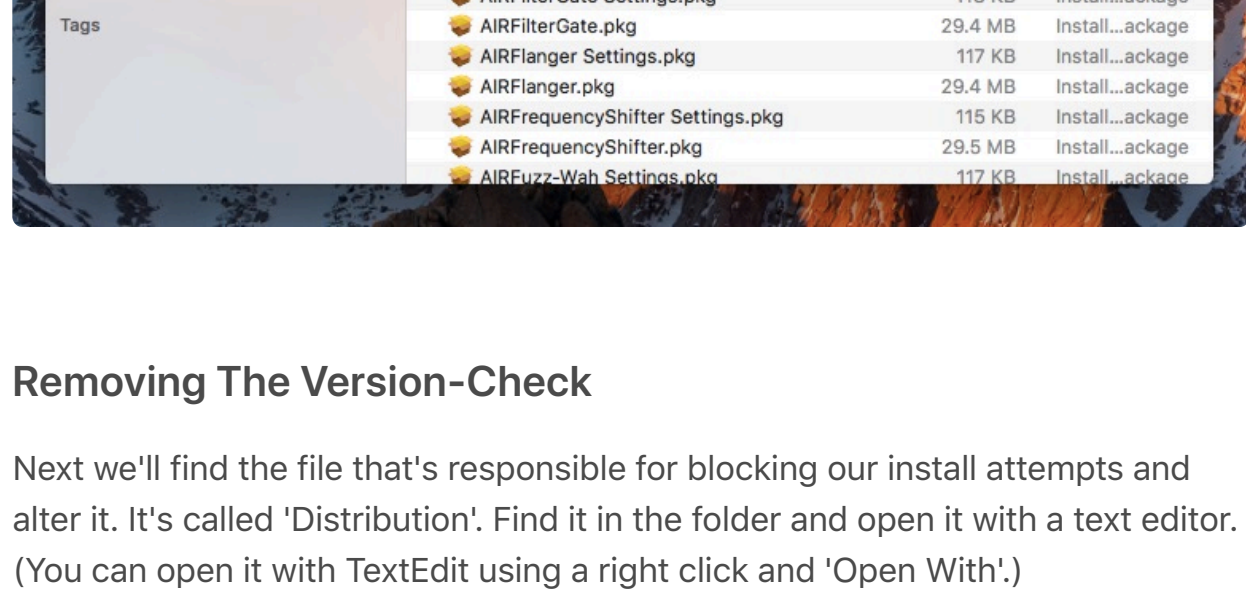
```
$ pkgutil --expand /Volumes/Pro\ Tools/Install\ Pro\ Tools.pkg
```

'pkgutil' also wants a name for the folder it will extract the contents to. Append 'package' to the command and finally hit enter. It might take a while to finish.

```
$ pkgutil --expand /Volumes/Pro\ Tools/Install\ Pro\ Tools.pkg pack
```

Inside the 'Installer' folder we created earlier is now a 'package' folder where the contents of the installer are. Open it in the Finder:

```
$ open package
```



Removing The Version-Check

Next we'll find the file that's responsible for blocking our install attempts and alter it. It's called 'Distribution'. Find it in the folder and open it with a text editor. (You can open it with TextEdit using a right click and 'Open With'.)

The file contains several functions which execute the checks. The one we're interested in is right near the top and called 'installationChecks'. The start of the function looks like this:

```
function installationChecks()
{
    var resultRAM = installationCheckRAM();
    var resultOS = volumeCheckLeopard();
    var resultPT = installationCheckPT();

    [...]
}
```

What we need to do is change the 'resultOS' variable to always be true. This can be done by replacing 'volumeCheckLeopard()' with 'true' as shown here:

```
function installationChecks()
{
    var resultRAM = installationCheckRAM();
    var resultOS = true;
    var resultPT = installationCheckPT();

    [...]
}
```

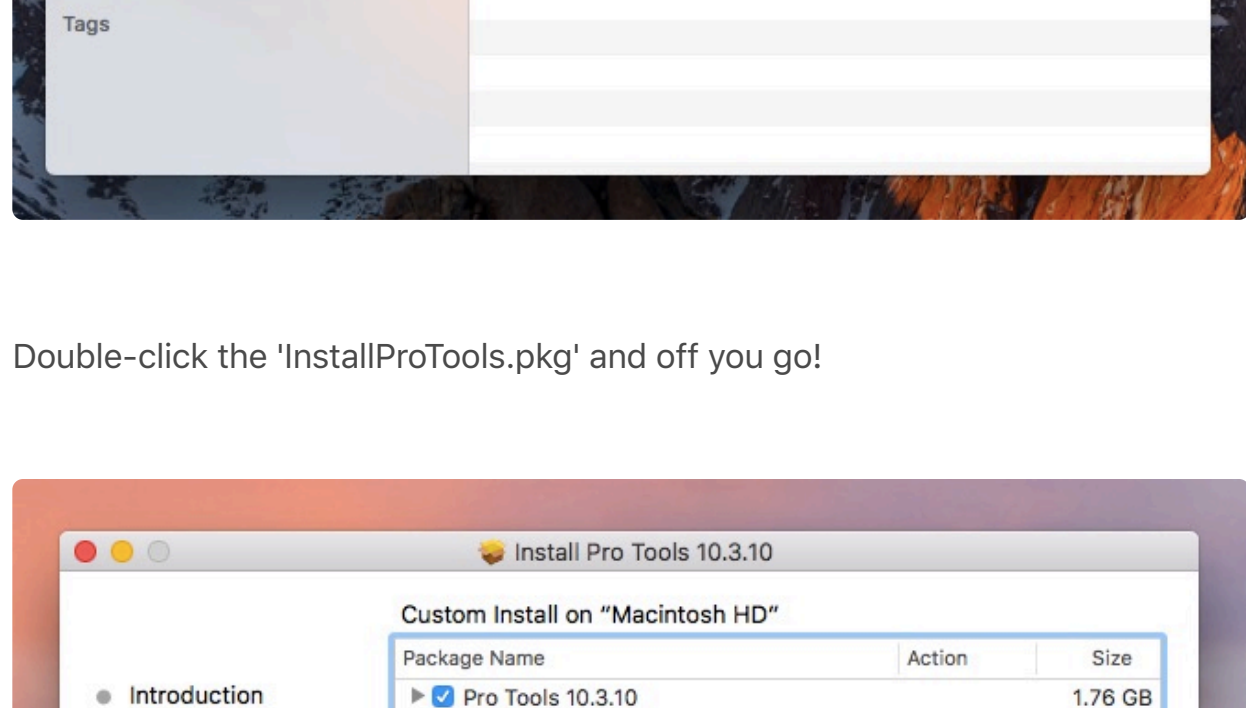
That's it. Save and close your editor. We now need to package all the extracted files back into an installer. Back in the terminal, tell 'pkgutil' to take the 'package' folder and turn it into a new installer. You can name the new installer what you want, we'll use 'InstallProTools.pkg' here. This operation can take a couple of minutes.

```
$ pkgutil --flatten package InstallProTools.pkg
```

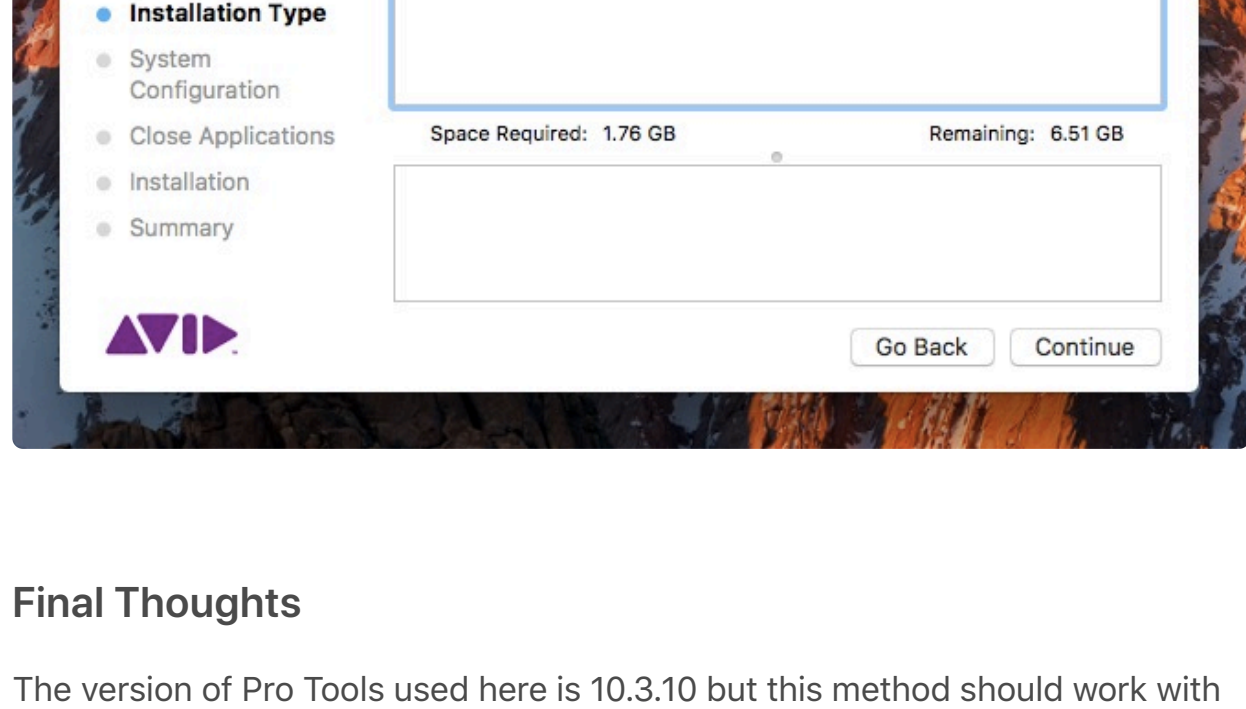
Finally, we need to copy the other files from your original .dmg or disk into the folder of our new installer. Open the folder in finder ...

```
$ open .
```

... and drag and drop the 'Uninstall Pro Tools', 'Additional Files' and 'Licenses' files from your .dmg or disk into the folder. It should look like shown here.



Double-click the 'InstallProTools.pkg' and off you go!



Final Thoughts

The version of Pro Tools used here is 10.3.10 but this method should work with slightly older versions like 10.3.9, too. I've been using Pro Tools 10 on macOS Sierra for couple of days now and didn't have any hiccups except minor graphical glitches. My guess is that Apple updated their Quartz graphics API and introduced some breaking changes which Pro Tools 10 suffers from. Core Audio however did not change much (if at all) during the last couple of OS updates and seems to run fine. Bouncing, effects etc are working for me. Finally, let me make clear that I am not the owner of an iLok dongle for Pro Tools 10. I cannot and will not test this guide with an illegally obtained copy of Pro Tools.