



**Roberto Panetta**



**Tension Software**

# Product Builder 2.3

## User Guide



We Make Software - Eclei.com

Product Builder © 2011-2019 Eclei - Roberto Panetta all rights reserved

Every effort has been made to ensure that the information in this manual is accurate.

Eclei is not responsible for printing or clerical errors.

Other company and product names mentioned herein are trademarks of their respective companies.

# Welcome to Product Builder

## Product Builder creates packages archive for the Apple Store

Product Builder creates .pkg archives, ready to be submitted to the App Store.

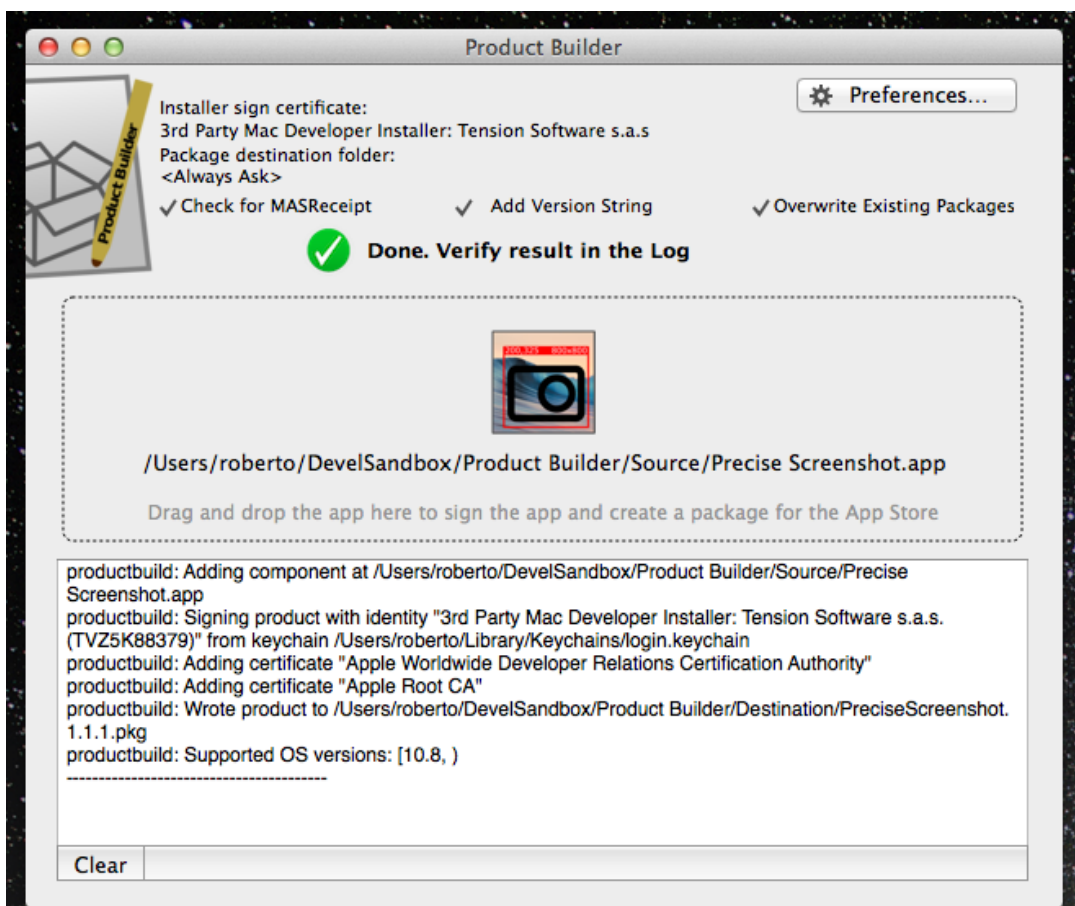
It allows to create packages in fast and error free way.

It allows to submit without problems applications with a name containing spaces inside as "My Wonderful App.app". This is not allowed from XCode.

Product Builder uses a simple interface.

Drop an app on the window of Product Builder and a package is created on your Hard Disk in the specified destination, ready to be submitted to the App Store.

The application provides also a log with the last activities, available to be inspected inside the main window.



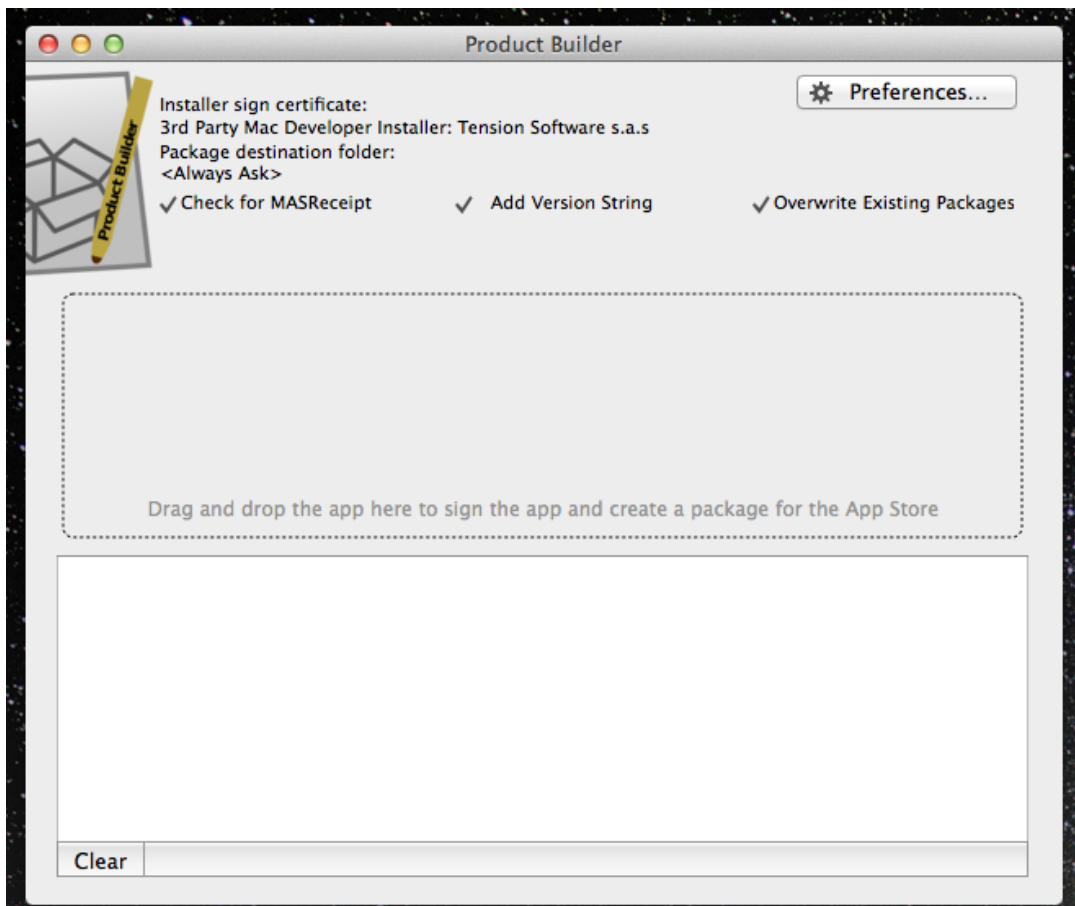
# Getting Started with Product Builder

## Using Product Builder is simple

You have already created an app ready to be submitted to the app store from your Hard Disk.

The app is already signed with your application certificate as Apple requires.

Launch Product Builder



Open the preferences and insert your installer certificate description (as inserted in your keychain)

You should have already obtained your certificates from Apple, inside Apple Developer Member Center as a registered software developer.

Insert this certificate, it will be used to sign the installer.

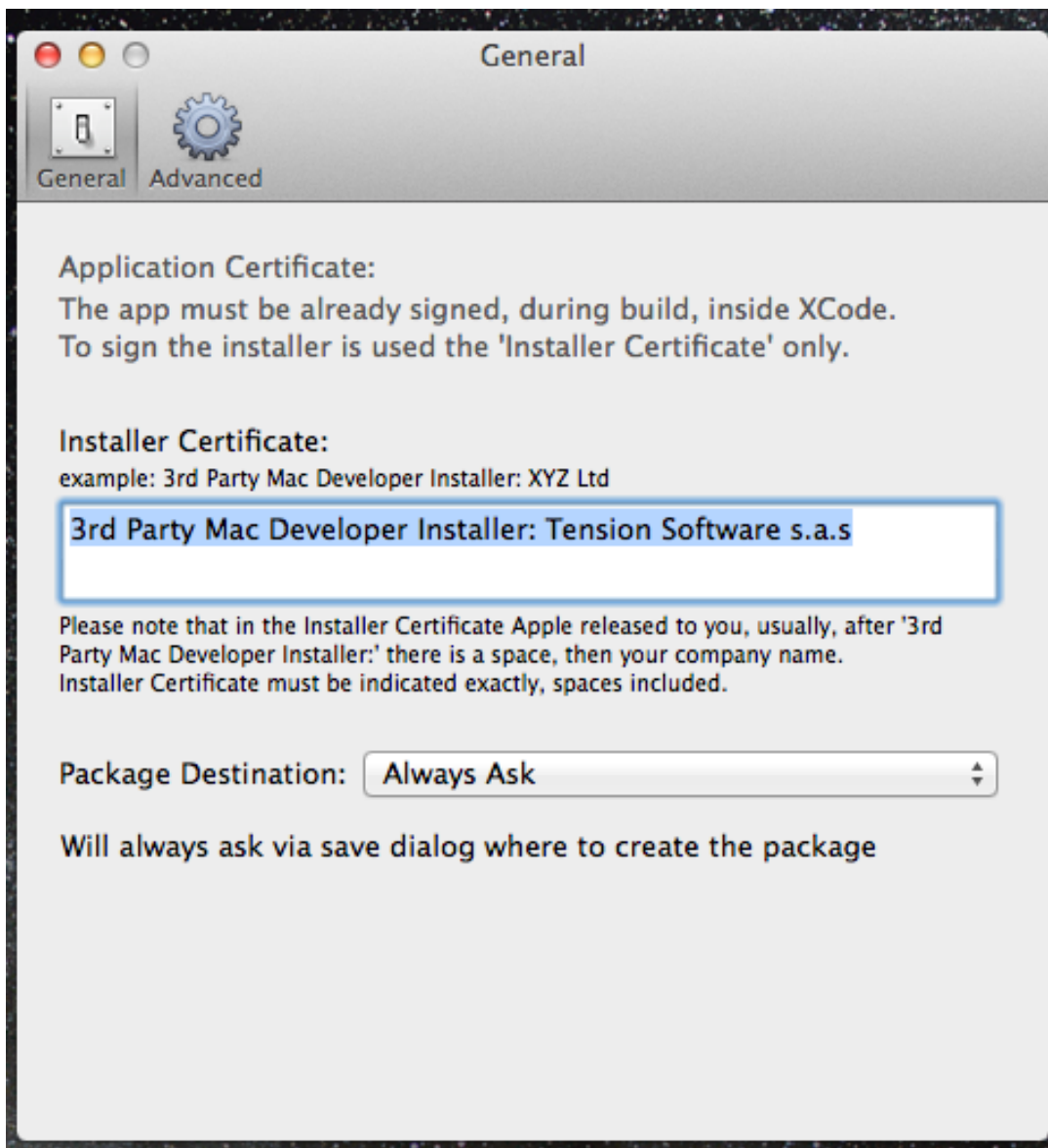
Usually it is something like

"3rd Party Mac Developer Installer: You Company Name"

Note that this certificate is different than the Application certificate you use to sign the application inside XCode. If you mix them things won't work.

then

Select to save the package in the same directory of the application



Close the preferences and drag the application you want use to create a package, on the drop area on the main window of Product Builder

You will be asked to confirm this destination at least one time, via a select dialog (you know, now we are all inside a sandbox and this explicit confirmation, is required by the system)

Product Builder will create the requested package in the specified destination, ready to be uploaded to the App Store.

# Reference

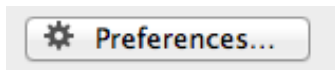
## Settings

Product Builder creates packages, ready to be submitted to the App Store, basing its works on settings specified in the preferences.

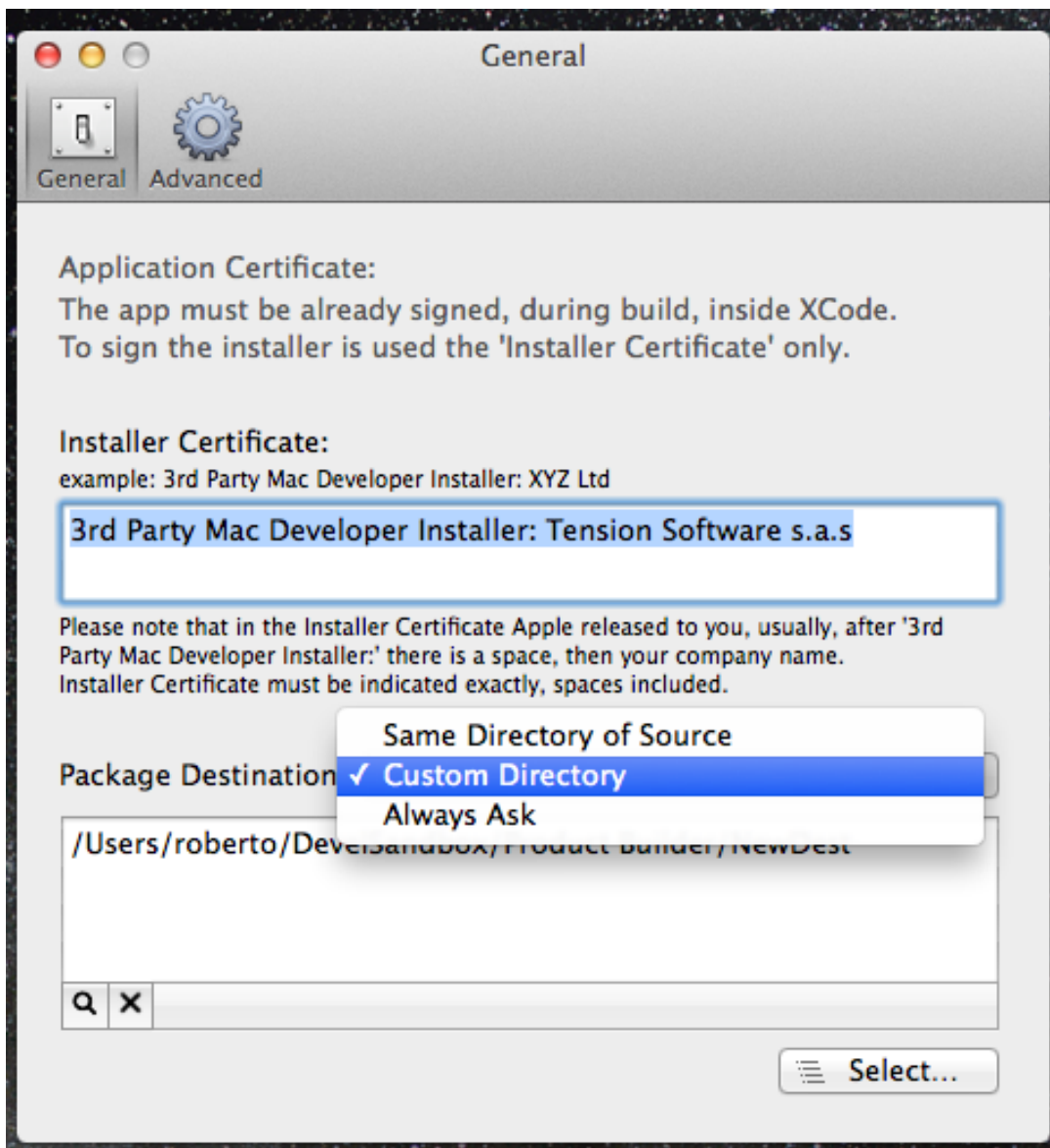
Product Builder is fully sandboxed and works in accordance with sandbox rules

Before creating your first package you need to insert the correct values in the preferences.

Open the preferences form the 'Product Builder' menu or clicking the 'Preferences' button in the main window



In the preferences dialog, in the 'General' section insert the name of your own certificates, as they are listed in your Keychain.



You should have already obtained your certificates from Apple, inside Apple Developer Member Center as a registered software developer.

The certificate has to be correctly installed inside your keychain.

Insert the certificate name to sign the installer (Installer Certificate) as is displayed inside your keychain.

Usually it is:

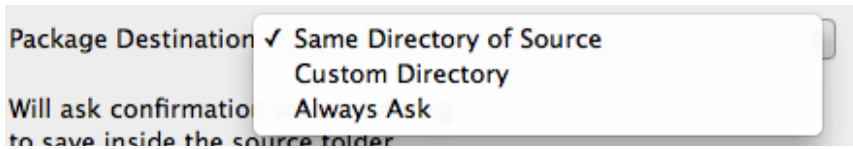
“3rd Party Mac Developer Installer: You Company Name”

(If you do not have your own certificates, you need to obtain them inside the Apple Developer area reserved to registered Apple developers.)

Specify where, on your hard disk, to create the packages.

You can select in the preferences to have the packages created

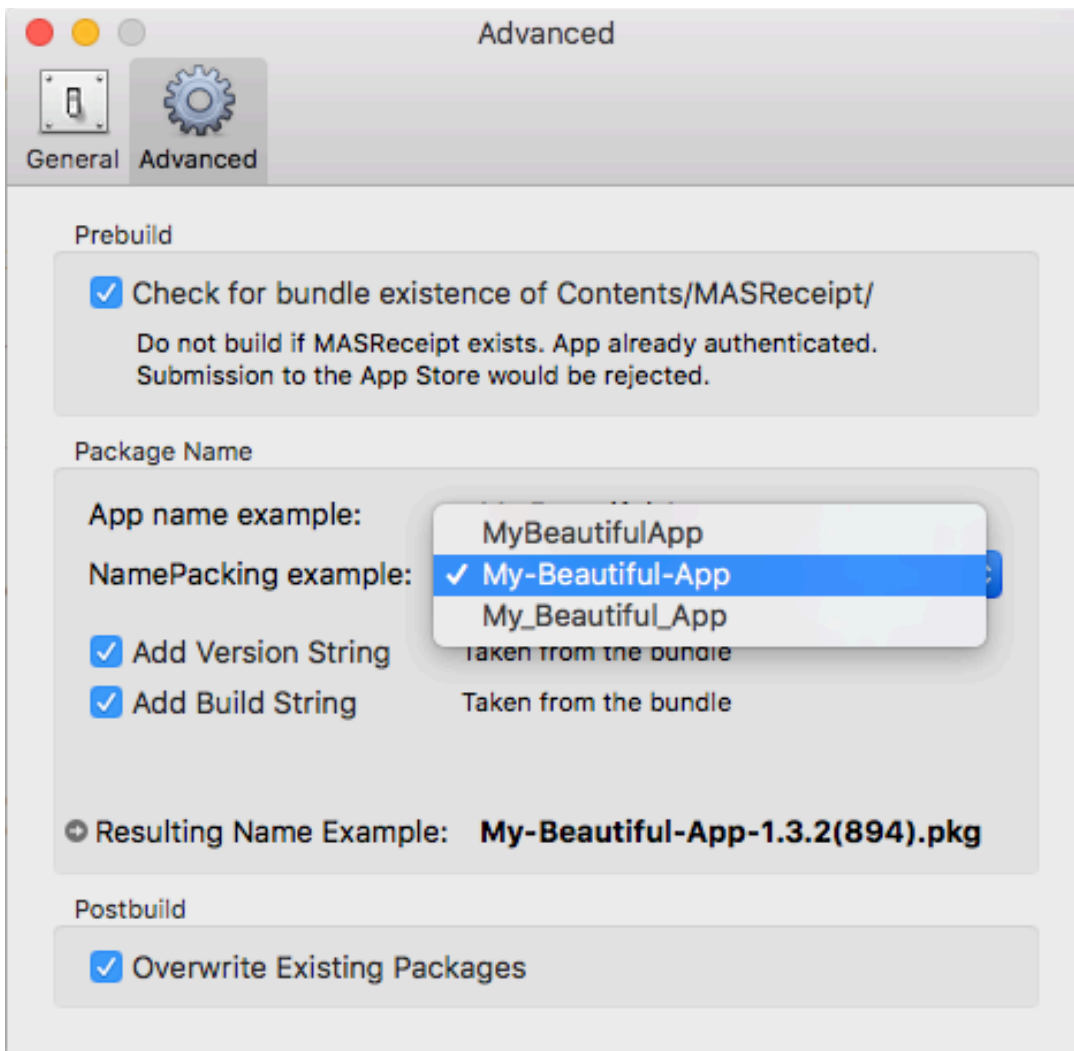
- 1) Near the application source (the one you will drop on the window)
- 2) In a specific destination using a popup control.
- 3) Always asking for destination.



The application will ask to confirm using a selection box the destination in accordance with the new rules for sandbox application.

When possible the app retain the location using a scoped bookmarks (a sandbox technology available from OS X 10.7.3 and later) of the destination folder and avoid to ask again to confirm the same location you previously confirmed. When the destination changes the you need to confirm again the destination to satisfy the sandbox requirements.

In the preferences, in the Advanced section you can specify:



In the Prebuild area (action to be executed before building):

To check inside the app you dragged on the window if a MASReceipt folder was created.

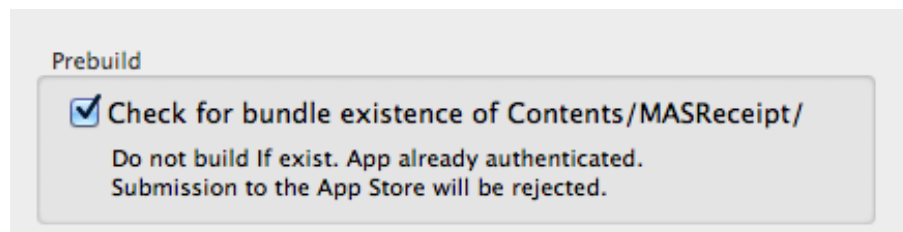
In case this will indicate you are creating a package from an app that was already authenticated with iTunes.

This is an error because you have to submit to the app store a 'clean' app, never authenticated with the store.

This is an error that can happen, because you, for test purpose, often authenticate yourself your application using your test account, ... then you submit it to the App Store and not a fresh version. Damn!

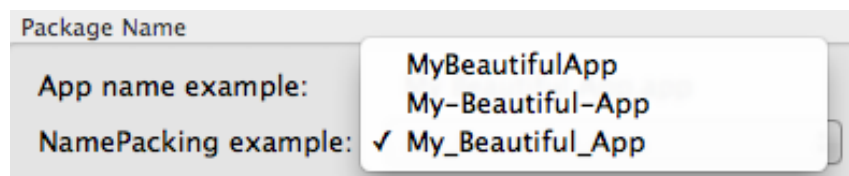
In that case Product Builder will rise an error, stopping the building and informing you.

You will avoid in that way to submit the wrong package.



In the Package name area:

How to adapt the name of the package to obtain a name without spaces inside (at the moment the App Store doesn't like filename with spaces inside) and XCode is not able to submit it.

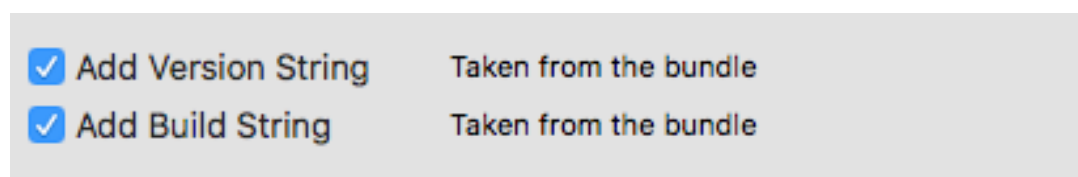


You can also select to have the release number of the app inserted at the end of the name or the build version or both.

They are read directly from the executable you dropped over Product builder.

This is a useful feature to keep all your submissions to the app store without the risk to submit an old package

You always know the release you are submitting via the Apple Application Loader because while you are selecting the package to submit you see the release number in the name of the file you select.





As example building the package to submit Product Builder I automatically obtained, ProductBuilder.2.2.pkg , just dropping the app on the Product Builder window, no input required. Ready to be submitted to the App Store. Product Builder knew what to do.

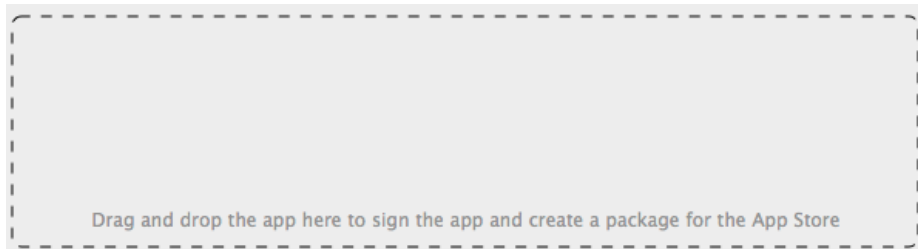
In the Postbuild area:

You can also select to overwrite, if present, in the selected destination, a package with the same name, or to create one with a name with a number appended by Product Builder itself.

## Creating the package

Creating the package is plain simple.

If the settings are correct, you have to drag and drop the application you want to create a package from, on the 'Product Builder' drop area.



The icon of the dropped application is shown in the drop area with the full path to the application

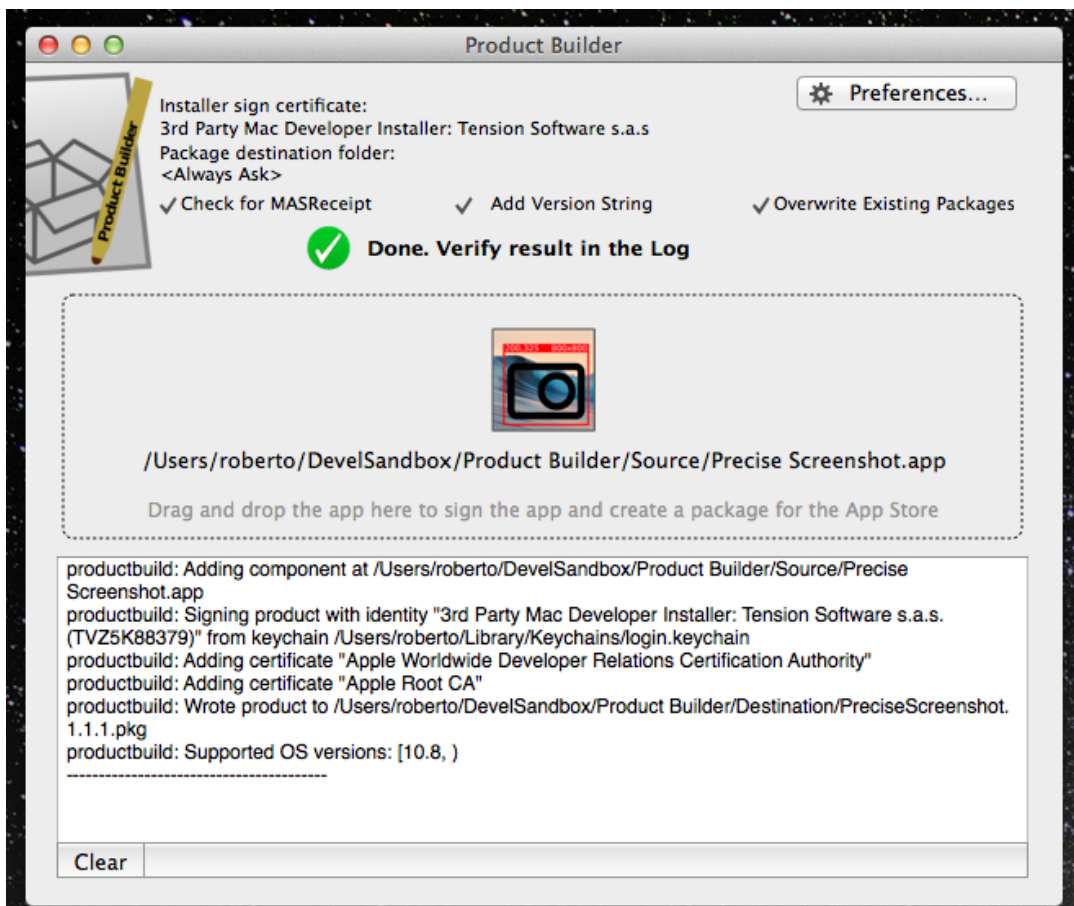
Open Dialogs may ask to specify the destination folder and the source parent folder (the one inclosing the source original app) because the system expects to have write permission on it even if it won't write here

A spinning indicator will indicate that Product Builder is working.

OS X may ask you the authorisation to access the certificates, confirm it.

After few seconds, it will indicate it created the package in the specified destination.

Done.



From version 2.0 of Product Builder the app used to create the package is left unmodified. It is not signed before creating the package

This requires the app was correctly signed before creating the package, inside XCode.

You have to correctly sign your app inside XCode as required.

In case read the Apple documentation on How to create an app for the App Store.

Product Builder sign only the installer package, ready to be transmitted to Apple for the submission to the Mac App Store

Product Builder just do one thing . In this case we think less is more and reducing the complexity reduces the possibility of troubles.

## Working under the Sandbox

Product Builder is a sandboxed app and need explicit permission to obtain write permission to folders

It needs permission to write to the destination folder and to the parent folder of the source (even if it does not write inside the later but the system expect to have it, so we had to add it to avoid an error)

The application will present two dialog, one asking to select the destination, unless you indicated it in the preferences and one asking to select the parent folder of the source.

The source is not modified by Product Builder.

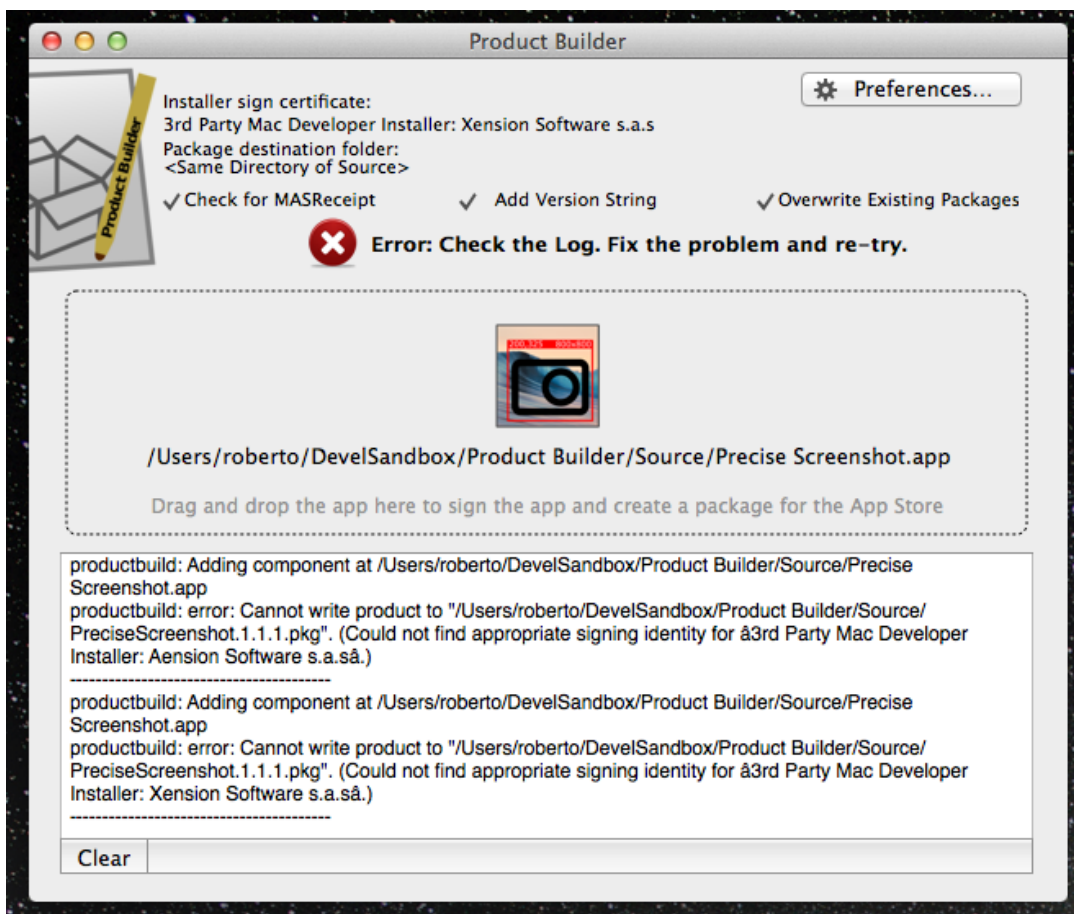
## Problems in Creating the Package

Problems, in development, always happens. Creating a package doesn't make an exception. Things can go wrong in many ways.

A certificate you inserted is misspelled, the destination folder is missing or other things...

Product Builder will inform you if something went wrong, as example if the requested certificate is not in found, showing a message error.

In some cases Product Builder can inform you in advance if some settings are not correct.



In any case it is a good practice to read the log to see exactly and in details how the package was created and if everything went in the right way. This is a developer tool, so the choice was to show all to the user of the software.

The package, if all went well, will be found at the specified location.

## Windows

Product Builder is based on the concept of 'Utility Windows'. Closing the windows quit the application exactly as selecting 'Quit' command from the menu

The windows remembers its position over different launches, in that way you can place it in the position you like for the next launch.

## Dark Appearance Mode

This application is also fully compatible with Dark Appearance Mode available on macOS from macOS 10.14 Mojave and successive.

## Help

Available under the Help menu also a PDF user guide.

## Support

You can also obtain support using the 'Email Support' command. An email will be prepared using your email client with the correct address to send to.  
Yes, we answer to your emails!

If you like Product Builder, please put a positive review on the App Store.  
Thank you.

## About Tension Software and Ecleti

All the software asset by Tension Software is now under Ecleti  
New company name but still the same owner and main developer (Roberto Panetta)  
So for the users nothing changes. We provide the same software and the services.  
All the user licenses are valid and assistance is provided as before to all our users.