Building an iOS signing key for PhoneGap in Windows

§

[←Back to listing](http://www.iandevlin.com/blog/)29 November 2012

One of the tasks I had to overcome when building a hybrid application that I’ve been working on was how to generate an iOS signing key that’s required by[PhoneGap Build](https://build.phonegap.com/) in order to generate iOS applications, on Windows. The steps to do this are outlined here.

Apple Developer Account

The one major thing that I will assume you have is an [Apple Developer](https://developer.apple.com/) account. There is no way around this, you *must* have one. Yes it costs $99 a year, but if you need to build an app for iOS, then you need to pay.

OpenSSL

You also need to download [OpenSSL](http://www.openssl.org/) and install it on your machine. This will end up somewhere like C:\OpenSSL-Win32 and it’s worth adding the bin directory to your Pathenvironment variable, which the steps below will assume you have done.

The steps

The first thing you need to do is generate a private key. Go to the command line and navigate to whatever directory you want to store the generated files in. Then type in the following to generate the key:

openssl genrsa -des3 -out ios.key 2048

This will generate a key and store it in the file ios.key. You will be prompted to generate a password for this key (once to create and once to verify), which you should take note of as you will require it later.

Next you need to generate a Certificate Signing Request or CSR file. You can do this by running the following command, which uses the ios.key file generated earlier:

openssl req -new -key ios.key -out ios.csr -subj "/emailAddress=MY-EMAIL-ADDRESS, CN=COMPANY-NAME, C=COUNTRY-CODE"

You will (obviously) need to change the email, company name and country code details to the correct values, e.g. -subj "/emailAddress=test@iandevlin.com, CN=iandevlin, C=DE". This will generate the required certificate signing request and store it in the fileios.csr.

Now you need to go to your [Apple Developer iOS Provisioning Portal](https://developer.apple.com/ios/manage/overview/index.action) in order to generate an iOS Development Certificate, using the ios.csr file you’ve just generated. Click on “Certificates” in the left hand side, and then “Request”. You will be prompted to upload a .csr file, and then wait for the certificate to be issued, which it will quite quickly, refresh the browser if you need to.

If you need any help with the portal, the website provides you with all you need to know.

Now download the development certificate that was issued and save it in the same directory where the other generated files are.

You now need to convert it to a PEM file which you can do with:

openssl x509 -in ios\_development.cer -inform DER -out ios\_development.pem -outform PEM

Where ios\_development.cer is the name of the development certificate created on the Apple Provisioning Portal and ios\_development.pem is the PEM file that we want to generate.

The penultimate file to generate is the P12 file, which uses both our private key (ios.key) and the iOS development certificate (ios\_development.pem):

openssl pkcs12 -export -inkey ios.key -in ios\_development.pem -out ios\_development.p12

You will be asked to enter the access phrase for the ios.key file (which you noted from earlier) and you will need to generate an export password for the P12 file and verify it. Theios\_development.p12 file is then generated.

The last file you need to generate is the provisioning profile, which again requires you to return to the Apple Provisioning Portal. There is plenty of documentation there on how to do this, so I won’t go into it here. Bear in mind that such certificates need to be tied to your iOS testing devices via their UDIDs, and again there is documentation on how to do this.

Once the provisioning profile is generated, download it (e.g.iOS\_Development.mobileprovision) and save it in the same place as the other files. This file will also need to be installed on each of your iOS testing devices.

You should now have everything that you need to generate an iOS signing key for PhoneGap Build:

* P12 certificate file
* provisioning profile
* certificate password

These steps can also be used to generate a distribution key for the iTunes Store.