# Installation on OSX

SUGGEST EDITS

Version 1.2, March 2017

## Change log

TABLE OF CONTE



03.22

- Added info on how to modify .bash_profile

Older

- Added hint where to find the path for cmake
- Make sure that your workspace path has *NO* spaces!!
- We now using the user's folder as installation target
- We added the mandatory step of adding Scade.app to Applications folder
- We clarified the Gradle requirement to not exceed 2.x

## Introduction - Up and running in 10 minutes

At SCADE, we are working on streamlining the installation over the next couple of versions. For now, we provide detailed documentation on how to setup your MAC for developing with SCADE. Average setup time is about 10 minutes...

All download links are included in the below instructions.

## Prerequisites

## Supported versions

SCADE 1.1 is supporting the following versions

- Swift 3.0
- iOS 9.0 and greater
- Android 5.0 and greater

## Requirements

- OS requirement : Swift, and therefore SCADE, requires OSX version 10.10 or higher
- XCode **8.x** is installed on the machine
- Android NDK **R11c**
- Android SDK **24.4.1** or higher
- CMake **3.1** or higher
- Gradle **> 2-14.1 but < 3.0**
- Java **8**
- Harddisk space: You need about 800 MB of harddisk space

# Step 1 - Install Android NDK

Download Android NDK from here
http://dl.google.com/android/repository/android-ndk-r11c-darwin-x86_64.zip

- Option 1 : You can either just use your browser for downloading OR
- Option 2 : curl **-O** <URL>
- Option 3 : wget <URL>

Once downloaded, lets say into the /Downloads directory, extract it to the /opt directory of your mac

- In your terminal session, goto the directory where the

file resides, for instance **/Downloads**

- `cd ~/<Directory where file is stored>` , ie. `cd ~/Downloads`

Unzip to user folder

- `unzip ./android-ndk-r11c-darwin-x86_64.zip -d ~/`

You should now find a directory `android-ndk-r11c` in your user directory.

## Step 2 - Install Android SDK

Download Android SDK from here https://dl.google.com/android/android-sdk_r24.4.1-macosx.zip

- Option 1 : You can either just use your browser for downloading,
- Option 2 : curl **-O** <URL>
- Option 3 : wget <URL>

Once downloaded, lets say into the /Downloads directory, extract it to the /opt directory of your mac

- In your terminal session, goto the directory where the file resides, for instance **/Downloads**
- `cd ~/<Directory where file is stored>` , ie. `cd ~/Downloads`
- `unzip ./android-sdk_r24.4.1-macosx.zip -d ~/`

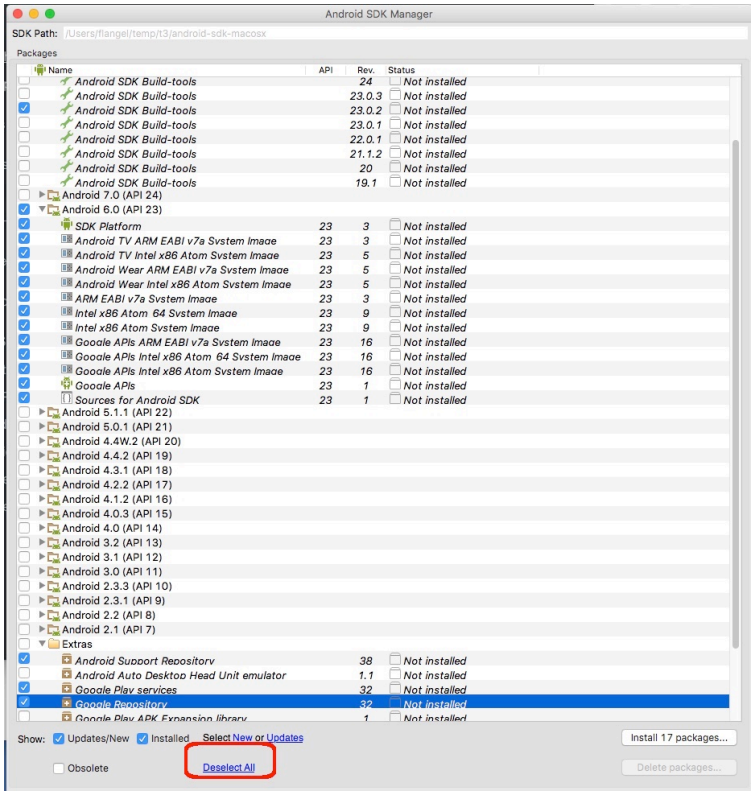
Your SDK is now located in the /<UserFolder>/android-sdk-macosx directory.

Once unzipped, we know need to configure Android using the Android SDK manager

- Start the Android SDK Manager from its home directory, for instance using `~/android-sdk-macosx/tools/android`

- If you have have Android SDK installed, it might be that the Android SDK can be found in this alternative directory `/Users/<Username>/Library/Android/sdk/tools` and you need to start android using `/Users/<Username>/Library/Android/sdk/tools/android`
- Uncheck all default selections and choose
  - Android 6.0 (API 23)
  - Tools / Android SDK Build-tools version 23.0.2
  - Extras / Android Support Repository
  - Extras / Google Play Services
  - Extras / Google Repository

  Your selection windows should look like this




- Now **continue by pressing** the Install 17 packages and **accepting the licenses** multiple

times

- Finally, let's clear the android folder
  - The **.android** folder. The .android folder that exists in the home directory of the user holdes settings that aren't needed and we want to start with clean slate.
  - Delete the android folder using `sudo rm -Rf ~/.android`

## Step 3 - Install Gradle

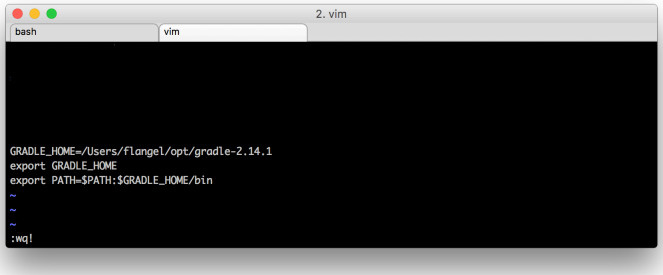
Gradle is our favourite build tool we use to build our Android and iOS app. If you are already using Gradle **>= 2.14.1 and < 3.0** , you can skip this step.

Download Gradle from here https://services.gradle.org/distributions/gradle-2.14.1-all.zip

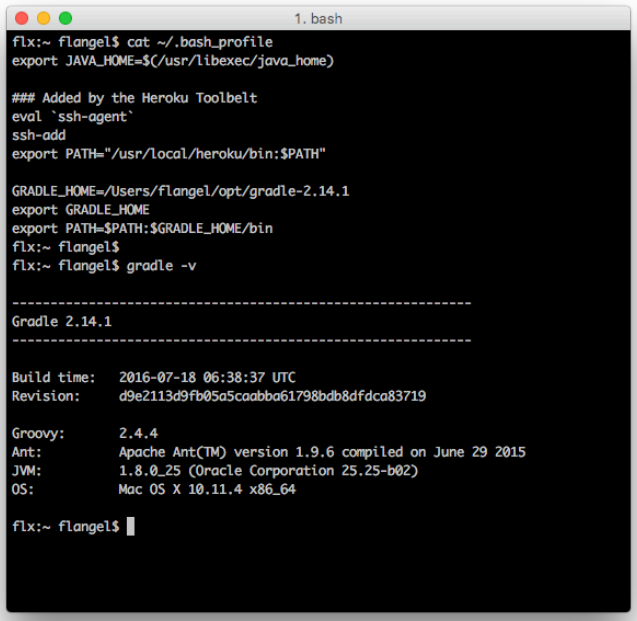
- Option 1 : You can either just use your browser for downloading,
- Option 2 : curl **-O** <URL>
- Option 3 : wget <URL>

Gradle can be installed in different locations. We use /usr/local/gradle

- Extract gradle `unzip gradle-2.14.1-all.zip -d /usr/local/gradle/`
- Export your home variable for gradle, GRADLE_HOME. Type `export GRADLE_HOME=/usr/local/gradle/gradle-2.14.1` in your terminal.
- Modify your .bash profile to make the changes permanent: Type `vi ~/.bash_profile`
- Edit the file and add the following




- Save file using Escape :wq!
- The final path you have to supply to SCADE is your home directory extended by /bin/gradle
- Your gradle -v to review your version Your bash file should look similar to this:




## Step 4 - Install CMake

CMake is used to compile and build the Apple iOS software

Option 1 : Use Homebrew (if you have homebrew on your Mac) to install cmake

- Type `brew install cmake` in your terminal
- Done. `cmake -version` should give you cmake message stating a version higher than 3.4.1

Option 2 : Download the dmg package

- You can download the dmg package here https://cmake.org/files/v3.6/cmake-3.6.2-Darwin-x86_64.dmg
- Execute your dmg

> **Location of cmake**
>
> - If you use homebrew, cmake is installed in /usr/local/bin/ and the setting in SCADE preferences needs to **/usr/local/bin/cmake**
> - If you use the dmg package option, the directory cmake is installed in is /Applications/CMake.app/Contents/bin and you need to set the SCADE setting accordingly to **/Applications/CMake.app/Contents/bin/cmake**

## Step 5 - Directory paths settings

SCADE needs to know the different paths where the software is installed. Depending on whether you choose different installation directories, your software should be found here:

| Software | Location |
|---|---|
| Android NDK | /Users/*<Username>*/android-ndk-r11c |
| Android SDK | /Users/*<Username>*/Library/Android/sdk |
| Gradle | /usr/local/gradle/gradle-2.14.1/**bin/gradle** |
| CMake | /usr/local/bin/cmake |
| CMake | /Applications/CMake.app/Contents/bin/cm (if you used dmg) |

| | |
|---|---|
| make (installed by default) | /usr/bin/make |

## Step 6 - Verify Swift 3.0 installation

Especially if you have multiple versions of Xcode installed, its critical that you are using the correct version. Doublecheck using these commands

- swift -version
- xcrun swift -version

All these commands should point to Swift 3.0:


## Step 7 - Install and run SCADE

*Download and extract SCADE from here*

Download SCADE here www.scade.io/download2

- Extract SCADE, i.e. `unzip scade1.0_beta01.zip -d ~/`

*Drop Scade.app icon into Appplications folder*

- One important new step, especially for Sierra users:

> **⚠ Drop Scade.app inside Applications folder**
>
> Please drop the Scade.app Icon into the Applications folder. By doing this, the app is not run in a sandbox and has all necessary permissions. Otherwise, you get all kind of weird issues when starting SCADE.

### *Start SCADE*

- Goto to the install directory `cd ~/Scade.app`
- Start binary `./Contents/MacOS/eclipse`
- ⚡ You might not directly launch SCADE by clicking on the icon in finder. That would result in an error message

### *Choose a good workspace location*

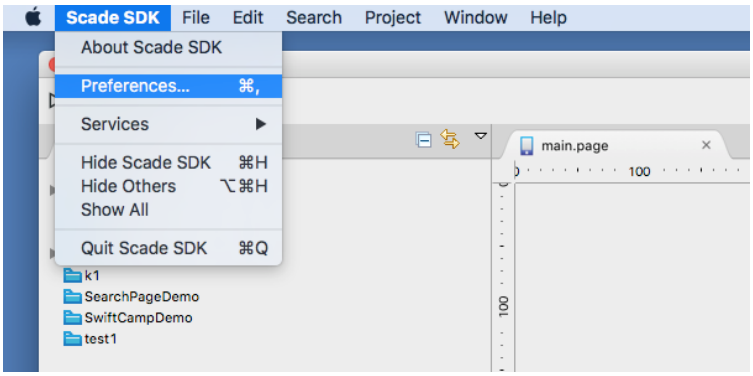
> **Workspace location**
>
> Please make sure that the workspace location you choose is OUTSIDE of the directory where SCADE resides. In this way, you can update the software without impacting your work.

> **Spaces in Workspace path**
>
> Currently, we don't support **spaces in the workspace** path. Please make sure that the workspace path has no spaces, otherwise the IDE crashes.
>
> Please also make sure that the workspace directory is not write protected, otherwise the IDE crashes as well.

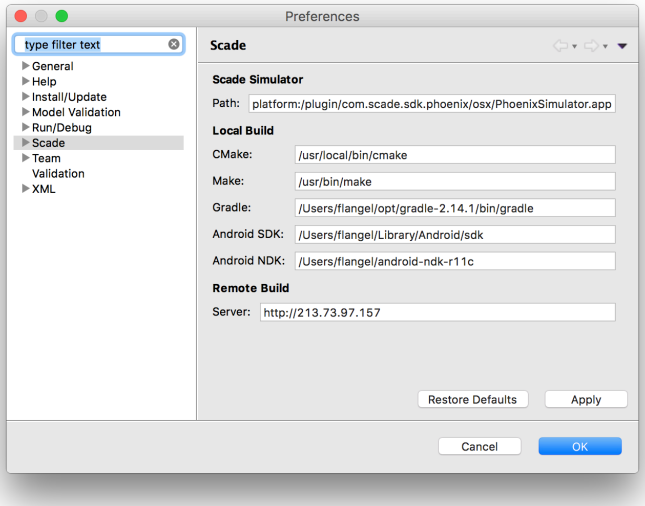
# Step 8 - Configure Android support

The following settings make it possible to compile to iOS and Android. Without the settings, you will be able to develop and run on iOS, but not be able to compile.

Open the preferences settings




and configure the paths. When you **press the apply button**, we do check the validity of the path.




Above an example of a valid configuration. Please make sure

- to append /bin/gradle to the home directory path of gradle

- the default Android SDK path on MAC is

  `/Users/<username>/Library/Android/sdk`

## Step 9 - Configure iOS support

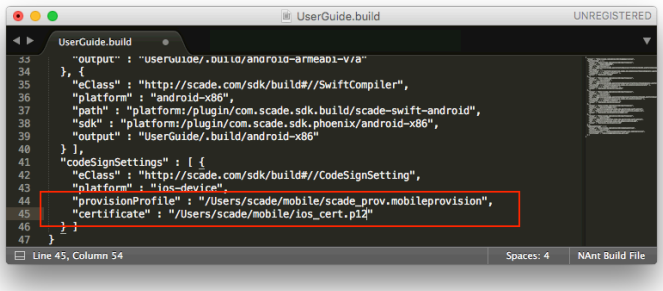
*Configure files*

Apple requires two different files for compiling apps

- a certificate file for signing the app, for instance **cert.p12**
- a mobile provision file **prov.mobileprovisioning**

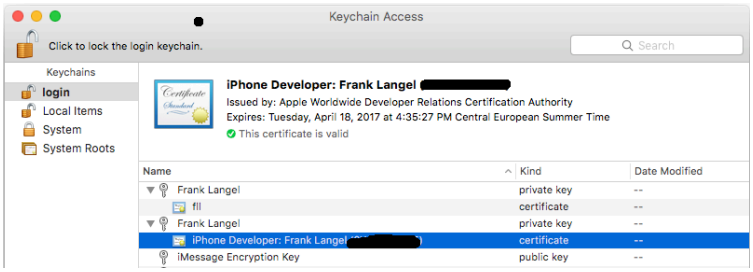
You need to specify the location of these files in the build file of your app.

1. Open the build file <YourApp>.build that can be found in the root directory of your project
2. Scroll to the codeSignSettings section at the end of the file
3. Modify the entries as shown below to reflect the location of your files




*Make sure certificate is in keychain*

The certificate needs to be part of the keychain. Double click on it and make sure its part of your keychain:




*Certificates and password support*

Certificates can be password protected. Currently, we don't support password protected certificates, but will add this feature in the future.

## Step 10 - Run SCADE

Congrats. You installed SCADE successfully. Over the next couple of months, we will provide a full fletched .dmg file with a nice installer to get you going even faster.

## Update SCADE

In order to update SCADE, please proceed as following

1. We assume you installed SCADE in a directory "Scade.app"
2. We assume you downloaded the new SCADE version

Please unzip the new version in another directory other than the root directory of "Scade.app". It is important that your directory with the new version is NOT called "Scade 2.app", but always "Scade.app"

3. Start SCADE from the new location.
   Unfortunately, currently you have to repeat the steps from step 8 **Configure Android support**.

We will improve SCADE to improve the ease of updating shortly.

## Troubleshooting tips

- none so far