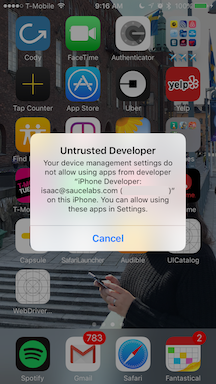
Development user setup

* Install Java Development Kit(JDK8) with respect to operating system and bitsize.
* Install Appium server command Line Interface.
  + goto [nodejs.org](http://nodejs.org/) and download nodejs and install it
  + npm install -g appium
* Install Eclipse IDE for Java Developers.
* Download and configure Maven from [maven.apache.org](http://maven.apache.org/)
* To Install maven plugin for Eclipse follow below steps
  + open Eclipse IDE
  + goto "help" menu
  + Install new software
  + click Add
  + Enter Name: "M2Eclipse"
  + Enter Location: [http://download.eclipse.org/technology/m2e/releases/](http://download.eclipse.org/technology/m2e/releases/1.4)
  + click ok
  + Select the checkbox "Maven Integration for Eclipse"
  + Finish the maven installation
  + Remove the Existing JRE from Build Path and Replace it with latest JRE Version
  + Change the Compiler to the latest version
* create bash\_profile and set environment variables
  + touch .bash\_profile
  + open .bash\_profile
  + set Environment variables as below
  + export JAVA\_HOME=$(/usr/libexec/java\_home)  
    export ANDROID\_HOME=/Users/mdayyala/Library/Android/sdk  
    export DEVICECONSOLE=/usr/local/lib/node\_modules

export M2\_HOME=/Applications/apache-maven-3.6.0  
export PATH="${PATH}:/$ANDROID\_HOME/platform-tools:/$ANDROID\_HOME/build-tools/28.0.3:/$ANDROID\_HOME/tools:/$JAVA\_HOME/bin:/$DEVICECONSOLE:/$ANDROID\_HOME/tools/bin:/$M2\_HOME/[bin: /usr/](http://bin/usr/)[bin:/usr/sbin:/bin:/sbin:/usr/local/bin:/usr/bin/xcrun:/Library/Developer/CommandLineTools/usr/bin](http://bin/usr/sbin:/bin:/sbin:/usr/local/bin:/usr/bin/xcrun:/Library/Developer/CommandLineTools/usr/bin)"

* **ANDROID setup**
* Download and Install Android SDK
* Open Eclipse IDE and set Environment variables
  + Goto Run Configurations-->Environment
  + name: ANDROID\_HOME,      [value: /Users/mdayyala/Library/Android/sdk](http://value/Users/mdayyala)
  + name: JAVA\_HOME,              [value: /Library/Java/JavaVirtualMachines/jdk1.8.0\_202.jdk/Contents/Home](http://value/opt/local/)
* **IOS setup**
* Download and Install Xcode and command line Tools from <https://developer.apple.com/download/more/>
* Open Eclipse IDE and set Environment variables
  + Goto Run Configurations-->Environment
  + name: HOME,      [value: /Users/mdayyala](http://value/Users/mdayyala)
  + name:PATH,         [value: /](http://value/Users/mdayyala)opt/local/bin:/opt/local/sbin:/usr/local/bin:/opt/apache-maven-3.6.0/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin://Users/mdayyala/Library/Android/sdk/platform-tools://Users/mdayyala/Library/Android/sdk/build-tools/28.0.3://Users/mdayyala/Library/Android/sdk/tools://Library/Java/JavaVirtualMachines/jdk1.8.0\_202.jdk/Contents/Home/bin://usr/local/lib/node\_modules://Users/mdayyala/Library/Android/sdk/tools/bin:/Applications/apache-maven-3.6.0/bin:/:/:/usr/bin:/usr/sbin:/bin:/sbin:/usr/local/bin:/usr/bin/xcrun:/Library/Developer/CommandLineTools/usr/bin
* Necessary installed software
* Install home-brew
  + ruby -e "$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/master/install>)"
  + brew doctor –> (verify your system is ready to brew)
* Install command line tools
  + xcode-select --install
* Install Appium
  + brew install node
  + npm install -g appium
  + npm install wd
  + appium &  →(To verify Appium is started)
* Install libimobiledevice
  1. brew install libimobiledevice --HEAD
  2. brew install carthage
  3. npm install -g ios-deploy
  4. gem install xcpretty
  5. brew install ideviceinstaller

If the WebDriverAgentRunneris successfully installed on the device, but in the Appium logs there is an error message then problem is that the developer is not trusted on the device. If you manually try to run the WebDriverAgentRunner app on the device, you will see a popup message:



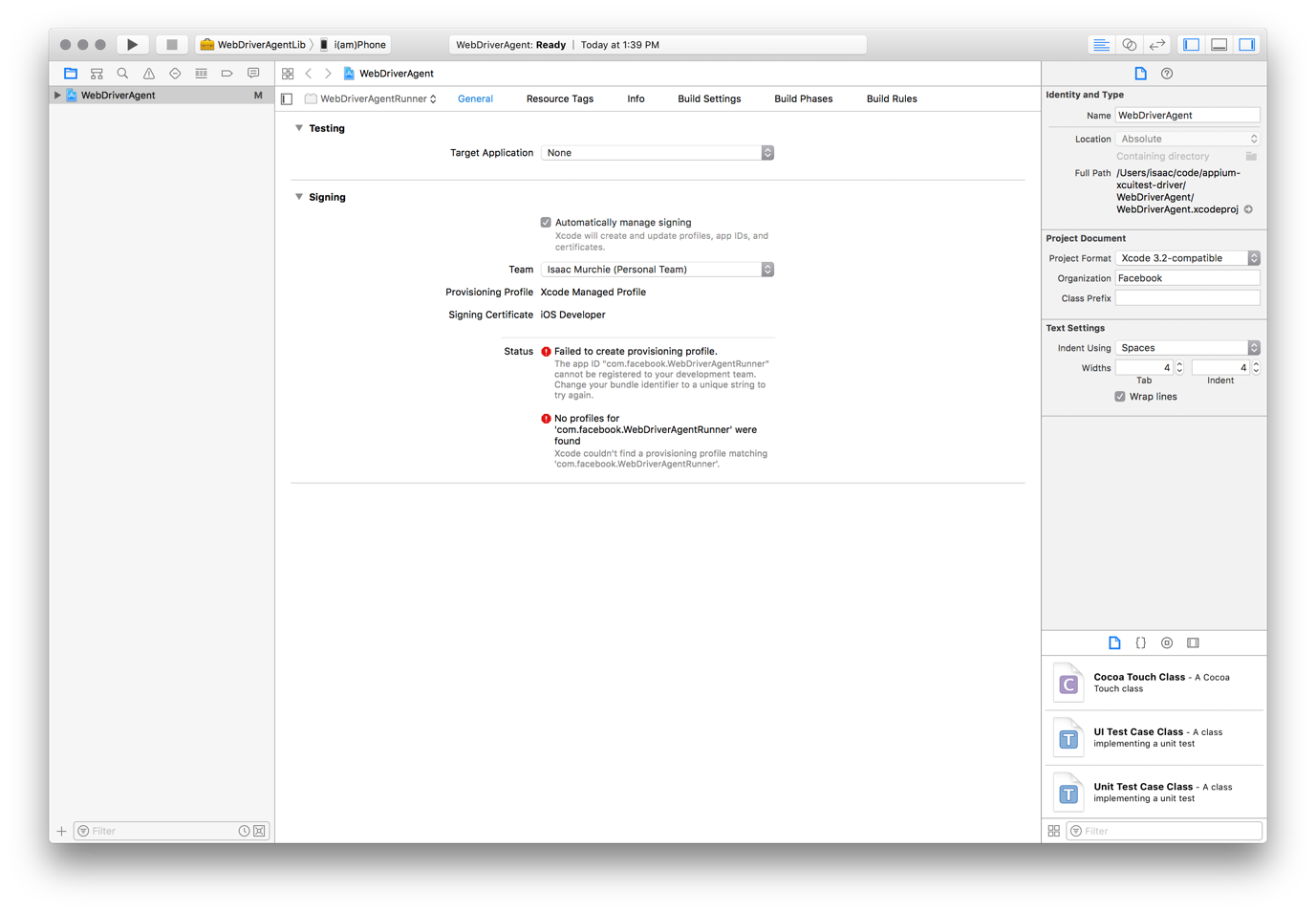
You can go to Settings => General => Device Management on the device to trust the developer and allow the WebDriverAgentRunnerapp to be run (see [Apple documentation for more information](https://support.apple.com/en-us/HT204460)).

* Goto the path "/usr⁩/local⁩/⁨lib⁩/node\_modules⁩/⁨appium⁩/⁨node\_modules⁩/appium-xcuitest-driver⁩/⁨WebDriverAgent⁩"
* open ⁨WebDriverAgent⁩.xcodeproj

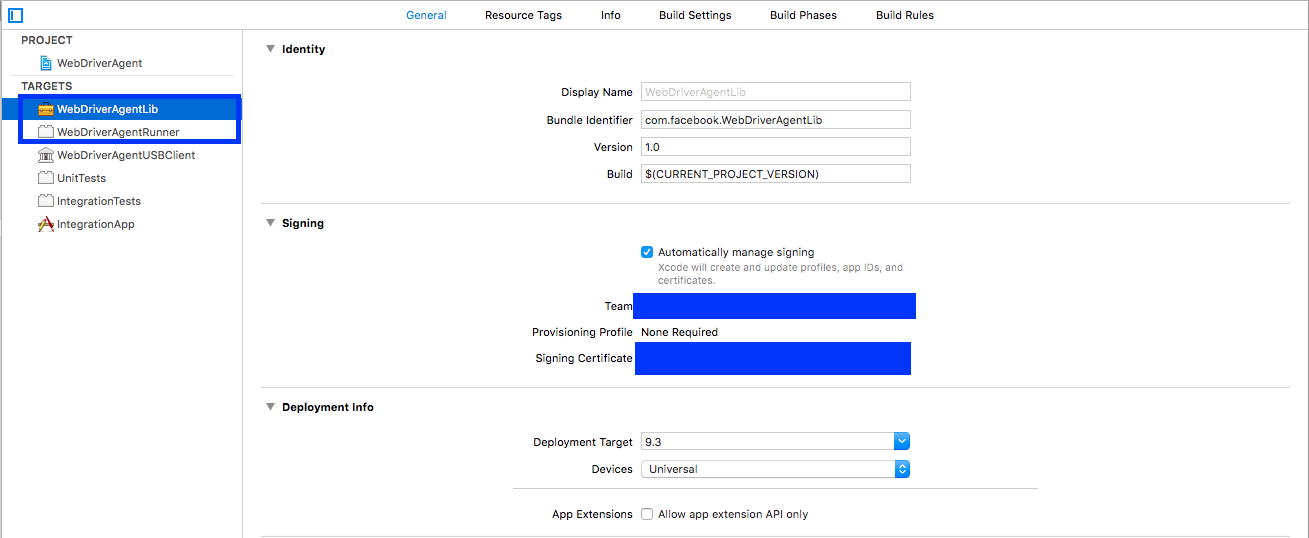
**Basic (manual) configuration**

Open WebDriverAgent.xcodeproj in Xcode. For **both** the WebDriverAgentLib and WebDriverAgentRunnertargets, select "Automatically manage signing" in the "General" tab,

This will manifest as something like an error that Xcode **failed to create provisioning profile**:



Goto signing → Team →Add an Account..

* Login with appleId
* select the Team name and click on "Download Manual Profiles"
* Then go back to "General" tab, and then select your Development Team. This should also auto select Signing Ceritificate. The outcome should look as shown below:
* 
* Alternatively, the provisioning profile can be manually associated with the project (keep in mind that this will have to be done each time the WebDriverAgent is updated and is *not* recommended):
* Find out where your Appium installation is:
  + $ which appium
* Given this installation location, /path/where/installed/bin/appium, WebDriverAgent will be found in **/usr/local/lib/node\_modules/appium/node\_modules/appium-xcuitest-driver/WebDriverAgent**. Open a terminal and go to that location, then run the following in order to set the project up:
  + mkdir -p Resources/WebDriverAgent.bundle
  + ./Scripts/bootstrap.sh -d
* Finally, you can verify that everything works. Build the project
  + xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination 'id=<udid>' test

xcodebuild build test -project /usr/local/lib/node\_modules/appium/node\_modules/appium-webdriveragent/WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination id='0ce01719b834e03b2989e3afdc844bbc1530118f' -configuration Debug -allowProvisioningUpdates DEVELOPMENT\_TEAM=KKAPNKP2N2

Steps followed to Setup the Run the Android Test Automation Suite in a fresh MAC Machine

1. Install Homebrew - <https://brew.sh/>
   1. Follow the commands to install Homebrew
2. Follow the instruction here to install jdk using Homebrew: <https://stackoverflow.com/questions/24342886/how-to-install-java-8-on-mac>
   1. brew tap caskroom/versions
   2. brew cask install adoptopenjdk8
3. Install nodejs from <http://nodejs.org/>
4. run the command npm install -g appium
   1. Command used to change the owner: sudo chown -R $USER /usr/local/lib/node\_modules
   2. If get the access issue, follow: <https://stackoverflow.com/questions/48910876/error-eacces-permission-denied-access-usr-local-lib-node-modules-react>
5. Download and install Eclipse
6. Follow the Steps in the Doco to install maven plugin
7. Download Binary tar.gz archive for Maven from <http://maven.apache.org/download.cgi>
8. Update the .bash\_profile by following the steps in docs
9. Download Android Studio for Mac from <https://developer.android.com/studio>
10. Open the Android Studio and Follow the process and install the required dependencies
11. Clone the repo into the local machine
12. In Eclipse, import the git project
13. In Eclipse, Right-click on the Project Folder, select the option Properties
14. Navigate to the Libraries Tab and update the JRE System Library to point the correct JDK

15. In Eclipse, Navigate to Run Configurations and open the Environments Tab

16. Add JAVA\_HOME and ANDROID\_HOME based on the Location of the Files Installed/copied

17. Navigate to the Test Tab in Run configurations and click on the Run button to run the Tests

CC: [Frank Ziegler](https://officeworks.atlassian.net/wiki/people/557058:18184170-75e9-4660-a08f-3cdb7fbb8ce8?ref=confluence)

Jenkins agent node setup

Headless user jenkinsagent

Make sure Java 1.8 is installed

git clone <https://sthoma40@bitbucket.officeworks.internal/scm/msd/ofw_ats_android.git>

cd ofw\_ats\_android/

mvn clean install

git checkout -b Automation/FixPageObjects

mvn test -Dtest="RunnerClassAndroid.java"

nano -w ~/.mavenrc

  export JAVA\_HOME=$(/usr/libexec/java\_home)   
  export ANDROID\_HOME=/Users/jenkinsagent/Library/Android  
  export ANDROID\_SDK\_ROOT=/Users/jenkinsagent/Library/Android

nano -w ~/.bash\_profile

  export JAVA\_HOME=$(java )  
  export ANDROID\_HOME=$HOME/Library/Android/sdk  
  export DEVICECONSOLE=/usr/local/lib/node\_modules  
  export M2\_HOME=/Applications/apache-maven-3.5.4  
  export PATH="${PATH}:/$ANDROID\_HOME/platform-tools:/$ANDROID\_HOME/build-tools/23.0.1:/$ANDROID\_HOME/tools:/$JAVA\_HOME/bin:/$DEVICECONSOLE:/$ANDROID\_HOME/tools/bin:/$M2\_HOME/[bin:/usr/bin:/usr/sbin:/bin:/sbin:/usr/local/bin:/usr/bin/xcrun:/Library/Developer/CommandLineTools/usr/bin](http://bin/usr/bin:/usr/sbin:/bin:/sbin:/usr/local/bin:/usr/bin/xcrun:/Library/Developer/CommandLineTools/usr/bin)"  
  export ANDROID\_SDK=$HOME/Library/Android/sdk  
  export PATH=$ANDROID\_SDK/emulator:$ANDROID\_SDK/tools:$PATH

curl <http://dl.google.com/android/android-sdk_r24.0.2-macosx.zip> -o android-sdk\_r24.0.2-macosx.zip

unzip android-sdk\_r24.0.2-macosx.zip

Android/sdkmanager  --update

sdkmanager  --update

sdkmanager --install "system-images;android\_28;google\_apis;x86"

download [devices.xml](https://officeworks.atlassian.net/wiki/download/attachments/857178273/devices.xml?version=1&modificationDate=1572573950484&cacheVersion=1&api=v2) into ~/.android/devices.xml

avdmanager create avd --force -n pixel3 --package "system-images;android-28;google\_apis;x86" --device "pixel\_3"

—>make sure antivirus will not block emulator

emulator @pixel3