KinomaJS Open Source Build

The KinomaJS Open Source Build can be built on MacOS X (Yosemite) and Linux 32-bit (Ubuntu 14.04). It requires Java version 7 or 8 and a number of readily available tools.

This document describes the setup and commands needed to build an application and simulator for KinomaJS.

A MacOS X (Yosemite) build host is necessary to build KinomaJS for MacOS, iOS and Android. There is an option to build an Xcode project to more easily develop for iOS and MacOS.

An Ubuntu 14.04 build host is necessary to build KinomaJS for the Linux GTK desktop and the Kinoma Create device.

Once the setup is done, building is pretty easy:

ant -Dtarget.platform=*target*

To build a debug version:

ant -Dtarget.platform=*target* -Dbuild.type=Debug

To build clean

ant -Dtarget.platform=*target* clean

Where *target* is one of:

- mac
- iphone/device
- iphone/simulator
- android
- linux/aspen (Kinoma Create)
- linux/gtk
- win

MacOS

For MacOS, Homebrew is employed to help manage the installation and maintenance of ant, cmake and Java.

1) Get a current version of Xcode from Apple's App Store

Make sure you run the Xcode app at least once so you can accept the license agreement and install updates.

2) Install Homebrew (http://brew.sh/)

```
$ ruby -e "$(curl -fsSL \
  https://raw.githubusercontent.com/Homebrew/install/master/install)"
$ brew install caskroom/cask/brew-cask
```

3) Install Java, ant and cmake

```
$ brew cask install java
$ brew install ant
$ brew install cmake
```

4) Get the tree

From a web browser:

http://github.org/kinoma/kinomajs

or from the command line:

```
git clone https://github.com/kinoma/kinomajs.git
```

5) Set up your environment

```
$ export F_HOME=/path/to/kinomajs
$ export XS_HOME=$F_HOME/xs
```

6) Build the app

```
$ cd $F_HOME
$ ant -Dtarget.platform=mac
```

7) Run the app

From the finder

Go to `$F_HOME/bin/mac/Release/balls/balls.app`

From the command line

```
$ open $F_HOME/bin/mac/Release/balls/balls.app
```

Versions:

XCode version 6.1.1
Java version 1.8.0_25
cmake 3.0.2
ant 1.9.4

iOS

The iOS build of KinomaJS is built on the MacOS Host set up as described above.

1) Install the MacOS host tools as described above.

2) Get the tree

From a web browser:

http://github.org/kinoma/kinomajs

or from the command line:

```
git clone https://github.com/kinoma/kinomajs.git
```

3) Set some environment variables

```
$ export F_HOME=/path/to/kinomajs
$ export XS_HOME=$F_HOME/xs
```

4) Build the app

```
$ cd $F_HOME
$ ant -Dtarget.platform=iphone/device
```

or build for the iPhone simulator

```
$ cd $F_HOME
$ ant -Dtarget.platform=iphone/simulator
```

5) The .ipa is found in

```
$F_HOME/bin/iphone/device/Release/balls/balls.ipa
```

If you have multiple developer accounts, you will be warned to include the a codesign switch to the command line. To identify what certificates you have installed, you can use:

```
$ security find-identity -v -p codesigning
```

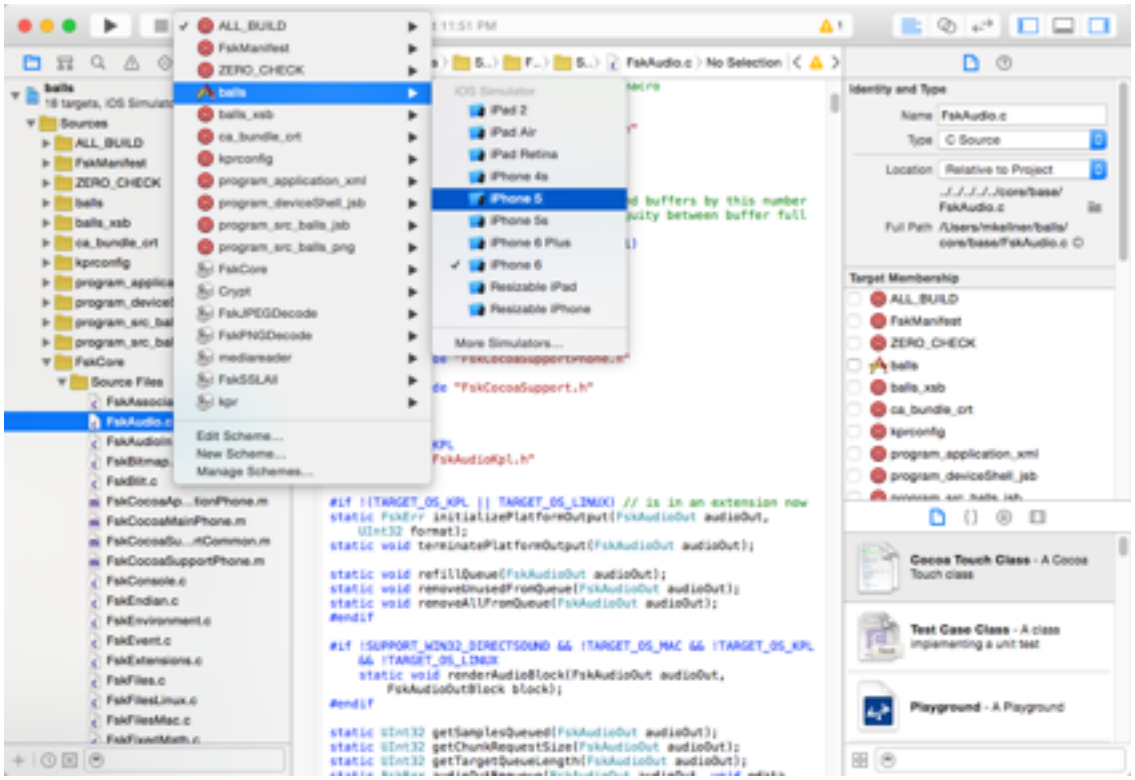
You would then add the switch **`-Dcodesign.id=`** to your build line:

```
$ ant -Dtarget.platform=iphone/device -Dcodesign.id=A290...
```

Xcode

If you add **`ide`** to the ant build line, an Xcode project will be created and Xcode launched. You can then edit and develop within the Xcode environment.

```
$ ant -Dtarget.platform=iphone/simulator ide
```



Choose the Application and particular simulator or iOS device.

Then hit the Play button. The iPhone simulator will launch with the balls app.

This will also work with the `mac` target.

```
$ ant -Dtarget.platform=mac ide
```

Android

The Android build of KinomaJS is built on the MacOS Host set up as described above. The Android NDK and SDK are also used.

1) Install the MacOS host tools as described above.

2) Install the Android SDK and NDK

a) If you do not have the Android tools already installed, you can use brew:

```
$ brew install android-sdk android-ndk
$ export ANDROID_NDK=/usr/local/opt/android-ndk
$ export ANDROID_SDK=/usr/local/opt/android-sdk
```

or

b) Install the Android SDK command line tools

http://developer.android.com/sdk/index.html#Other

Install as specified (as of Feb 2015)

```
$ tar -zxvf android-sdk_r24.0.2-macosx.zip
$ mv android-sdk-macosx /your/buildtools/path
$ export ANDROID_SDK=/your/buildtools/path/android-sdk-macosx
```

Install the Android NDK

http://developer.android.com/tools/sdk/ndk/index.html

Install as specified (as of Feb 2015)

```
$ chmod a+x android-ndk-r10d-darwin-x86_64.bin
$ ./android-ndk-r10d-darwin-x86_64.bin
$ mv android-ndk-r10d /your/buildtools/path
$ export ANDROID_NDK=/your/buildtools/path/android-ndk-r10d
```

3) Update your android tools and get platform-tools

```
$ android update sdk -u -t platform-tools
$ android update sdk -u -t android-17
$ android update sdk -u -t \
      $(android list sdk -e | grep build-tools | \
      sed -ne 's/.*"\(.*\)".*/\1/p' | head -1)
```

4) Get the tree

From a web browser:

http://github.org/kinoma/kinomajs

or from the command line:

```
git clone https://github.com/kinoma/kinomajs.git
```

5) Set some environment variables

```
$ export F_HOME=/path/to/kinomajs
$ export XS_HOME=$F_HOME/xs
```

6) Build the app

```
$ cd $F_HOME
$ ant -Dtarget.platform=android
```

7) The .apk is found in

```
$F_HOME/bin/android/Release/balls/balls.apk
```

Versions:

Android NDK - stable r10c
Android SDK - stable 23.0.2

Linux

The Linux builds of KinomaJS are built on a 32-bit Ubuntu 14.04 (or equivalent). We have had success with Ubuntu on VirtualBox (on MacOS).

Linux/gtk

1) Install a Java7 or Java8

2) Install the developer tools, ant and cmake

```
$ sudo apt-get install build-essential checkinstall
$ sudo apt-get install -y libgtk-3-dev libasound2-dev zlib1g-dev

$ sudo apt-get install ant
$ sudo apt-get install cmake
```

3) Get the tree

From a web browser:

http://github.org/kinoma/kinomajs

or from the command line:

```
git clone https://github.com/kinoma/kinomajs.git
```

4) Set some environment variables

```
$ export F_HOME=/path/to/kinomajs
$ export XS_HOME=$F_HOME/xs
```

5) Build the app

```
$ cd $F_HOME
$ ant -Dtarget.platform=linux/gtk
```

6) Run the app

The binary is found in

```
$F_HOME/bin/linux/gtk/Release/balls/balls
```

Kinoma Create

Native apps built for Kinoma Create are built on a Linux host machine.
Please set up your host machine as described above.
The Kinoma Create is known as the linux/aspen target platform.

Linux/aspen

1) Set up the Linux host as described above.

2) Install the aspen toolchain and sysroot

```
$ mkdir /your/buildtools/path
$ cd /your/buildtools/path
$ wget http://downloads.kinoma.com/aspen/sysroot.tbz
$ tar -jxvf sysroot.tbz
$ wget http://downloads.kinoma.com/aspen/toolchain.tbz
$ tar -jxvf toolchain.tbz
```

3) Get the tree

From a web browser:

http://github.org/kinoma/kinomajs

or from the command line:

```
git clone https://github.com/kinoma/kinomajs.git
```

4) Set some environment variables and your PATH

```
$ export F_HOME=/path/to/kinomajs
$ export XS_HOME=$F_HOME/xs
$ export ARM_MARVELL_LINUX_GNUEABI=/your/buildtools/path/arm-marvell-
linux-gnueabi
$ export FSK_SYSROOT_LIB=/your/buildtools/path/arm-bin
```

5) Build the app

```
$ cd $F_HOME
$ ant -Dtarget.platform=linux/aspen
```

6) Run the app

The package is found in

```
$F_HOME/bin/linux/aspen/Release/balls.tgz
```

Transfer the .tgz file to your device and decompress.

The executable will be at balls/balls

Windows

The Windows builds of KinomaJS are built with the Express (free) version of Visual Studio. It has been successfully built with Windows 7, Windows 8, Visual Studio 2010 Express and Visual Studio 2013 Express.

1) Install a Java7 or Java8

2) Install the developer tools (Visual Studio Express), ant and cmake

https://www.visualstudio.com/downloads/download-visual-studio-vs
http://ant.apache.org/
http://www.cmake.org

3) Get the tree

From a web browser:

http://github.org/kinoma/kinomajs

or from the command line:

```
git clone https://github.com/kinoma/kinomajs.git
```

4) Set or verify some environment variables

Start->Control Panel->System->Advanced system settings->Advanced->Environment Variables...

```
ANT_HOME=\path\to\ant    (example: ANT_HOME=c:\Users\me\apache-ant-1.9.4)
F_HOME=\path\to\kinomajs       (example: F_HOME=c:\Users\me\kinomajs)
XS_HOME=\path\to\kinomajs\xs  (example: XS_HOME=c:\Users\me\kinomajs\xs)
```

5) Launch a Visual Studio Command Prompt

Start->All Programs->Visual Studio Command Prompt (2013)

6) Build the app

```
$ cd %F_HOME%
$ ant -Dtarget.platform=win
```

7) Run the app

The binary is found in

```
%F_HOME%\bin\win\Release\balls\balls.exe
```