iContinuousIntegration

Oleksandr Dodatko

What's Covered

Managing shared projects with xCode

Building a project without xCode GUI

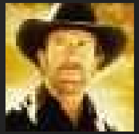
Creating “universal binary” libraries and frameworks

Deploying projects and libraries for QA

More Fun for Developers

Unit testing with GHUnit

Using Hudson build server

(it has a Chuck Norris plug-in) 

Running applications on simulator without xCode

A Build server should

Checkout project sources

Run a build script

Deploy product archives

Publish test reports

A Build Script Should

Build main products

Create *.ipa packages for main products

Run clang static analyzer

Build unit tests

Run unit tests with iphonesim

Package *.ipa and *.app entries to *.zip archive

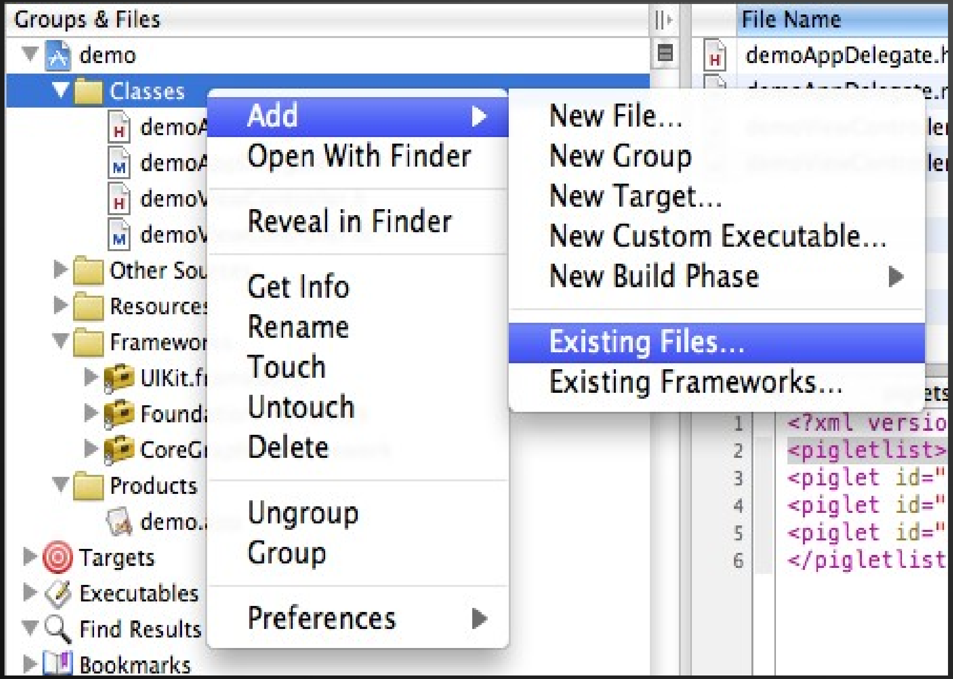
Prepare unit test and clang reports for deployment

Hudson CI quick start

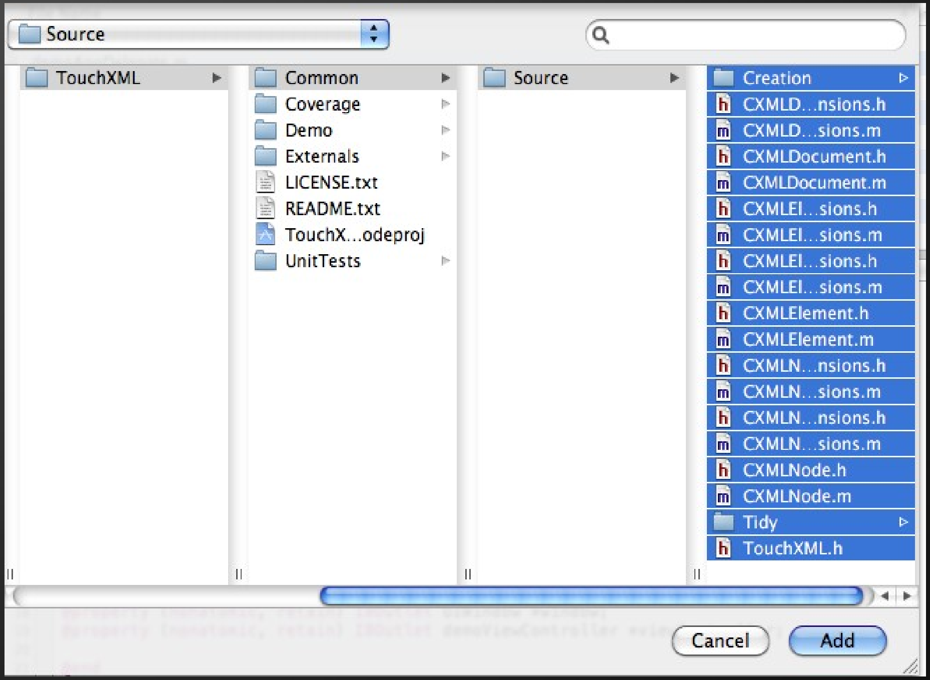
SICCI for Xcode Plugin – sicci_for_xcode
Clang scan-build plug-in – clang-scanbuild-plugin
Testflight Plugin – testflight

Pros	Cons
Simple learning curve Easy to use	Scripting provides more control and flexibility

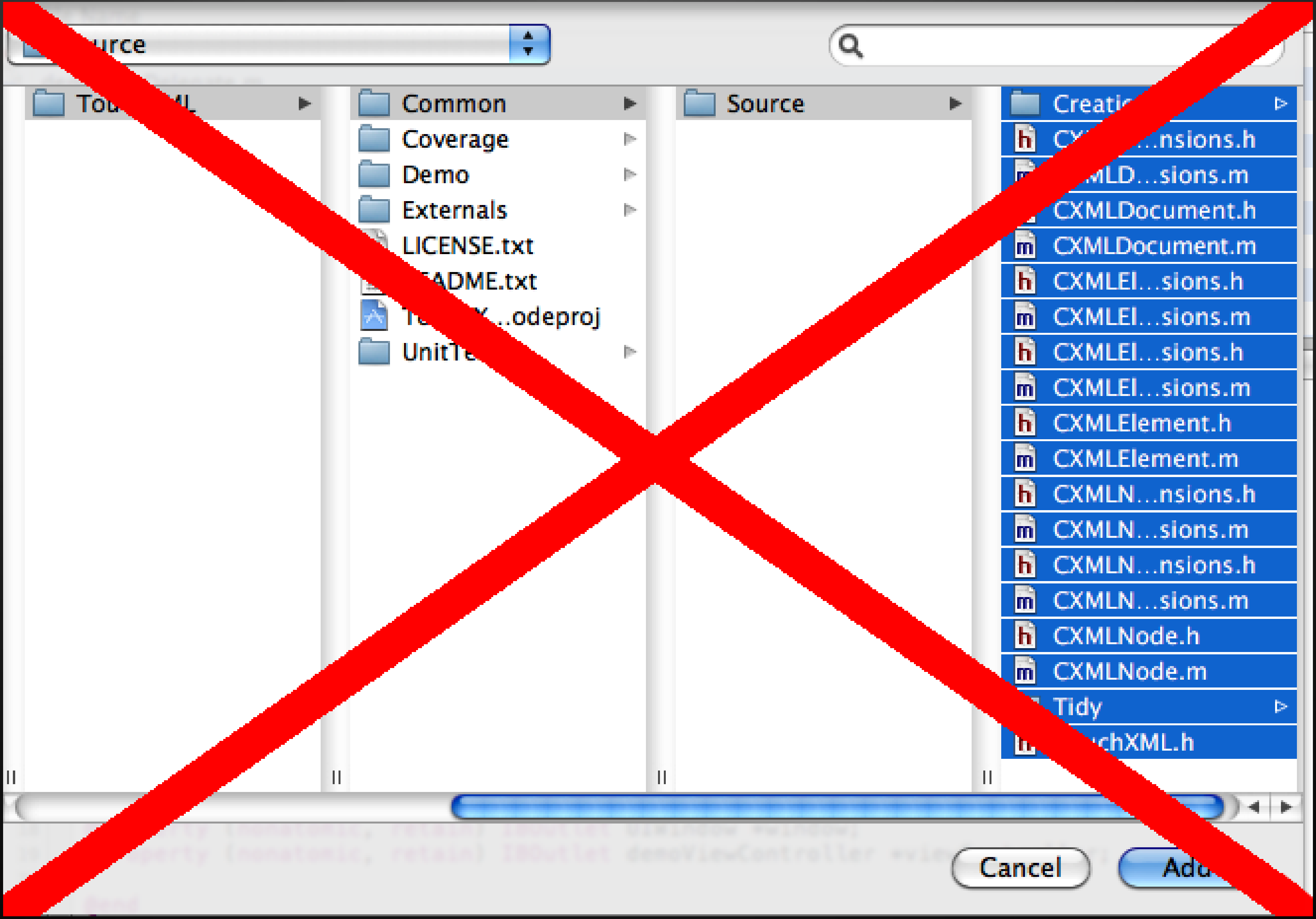
“Commonly Used” Project Organization



```
<?xml versio
<pigletlist>
<piglet id="
<piglet id="
<piglet id="
</pigletlist
```

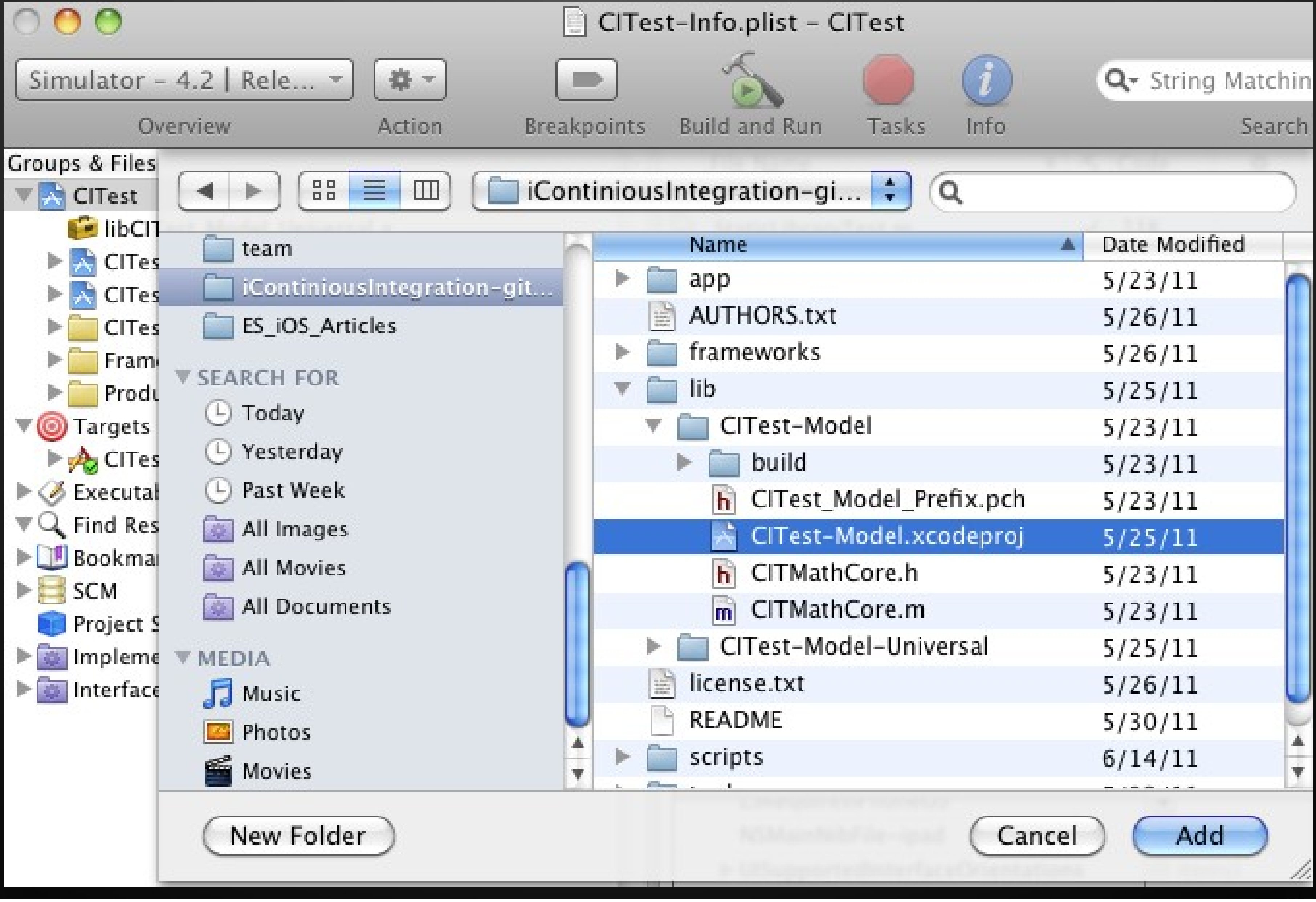


Wrong !!!


Source
TouchXML
Common
Coverage
Demo
Externals
LICENSE.txt
README.txt
TouchXML.xcodeproj
UnitTests
Source
Creation
CXML…nsions.h
CXMLD…sions.m
CXMLDocument.h
CXMLDocument.m
CXMLEl…sions.h
CXMLEl…sions.m
CXMLEl…sions.h
CXMLEl…sions.m
CXMLElement.h
CXMLElement.m
CXMLN…nsions.h
CXMLN…sions.m
CXMLN…nsions.h
CXMLN…sions.m
CXMLNode.h
CXMLNode.m
Tidy
TouchXML.h
Cancel
Add

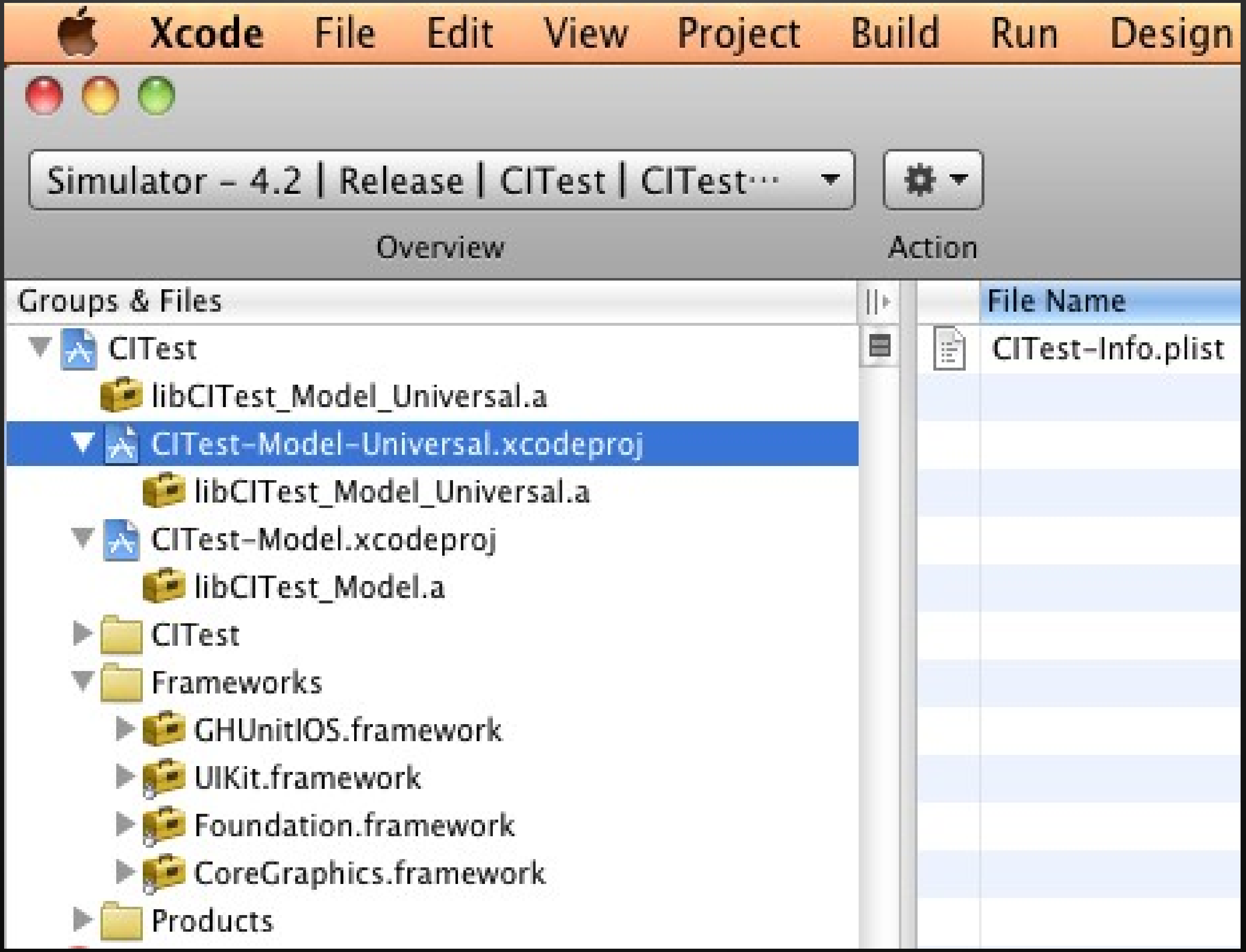
ONE Product, ONE XCODE PROJECT

Library Project How-To



Name	Date Modified
app	5/23/11
AUTHORS.txt	5/26/11
frameworks	5/26/11
lib	5/25/11
CITest-Model	5/23/11
build	5/23/11
CITest_Model_Prefix.pch	5/23/11
CITest-Model.xcodeproj	5/25/11
CITMathCore.h	5/23/11
CITMathCore.m	5/23/11
CITest-Model-Universal	5/25/11
license.txt	5/26/11
README	5/30/11
scripts	6/14/11

Library Project How-To

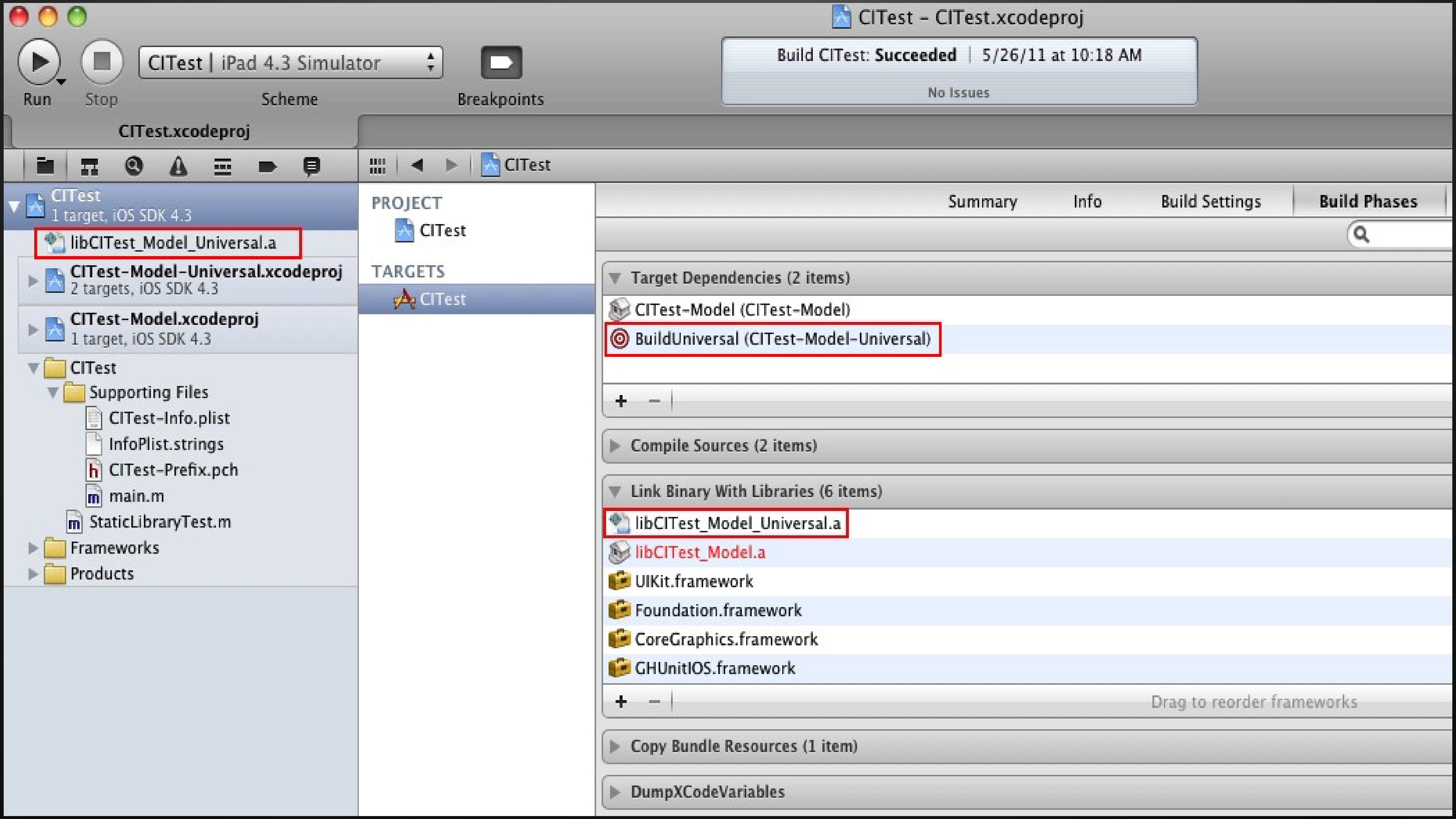


Simulator – 4.2 | Release | CITest | CITest…

CITest
libCITest_Model_Universal.a
CITest-Model-Universal.xcodeproj
libCITest_Model_Universal.a
CITest-Model.xcodeproj
libCITest_Model.a
CITest
Frameworks
GHUnitIOS.framework
UIKit.framework
Foundation.framework
CoreGraphics.framework
Products

CITest-Info.plist

Setting up Dependencies



CITest – CITest.xcodeproj

Build CITest: **Succeeded** | 5/26/11 at 10:18 AM

No Issues

- CITest — 1 target, iOS SDK 4.3
  - libCITest_Model_Universal.a
  - CITest-Model-Universal.xcodeproj — 2 targets, iOS SDK 4.3
  - CITest-Model.xcodeproj — 1 target, iOS SDK 4.3
  - CITest
    - Supporting Files
      - CITest-Info.plist
      - InfoPlist.strings
      - CITest-Prefix.pch
      - main.m
    - StaticLibraryTest.m
  - Frameworks
  - Products

PROJECT
- CITest

TARGETS
- CITest

Summary | Info | Build Settings | **Build Phases**

Target Dependencies (2 items)
- CITest-Model (CITest-Model)
- BuildUniversal (CITest-Model-Universal)

Compile Sources (2 items)

Link Binary With Libraries (6 items)
- libCITest_Model_Universal.a
- libCITest_Model.a
- UIKit.framework
- Foundation.framework
- CoreGraphics.framework
- GHUnitIOS.framework

Drag to reorder frameworks

Copy Bundle Resources (1 item)

DumpXCodeVariables

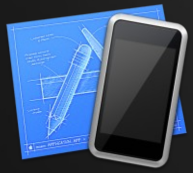
Creating Universal Binaries

1. No need to open your source code.
2. A better experience for library users

1. Build a library version for the device.
2. Build a library version for the simulator.
3. Combine them to a single binary
4. Deploy universal library to the “frameworks” directory.

Combining Binaries

```
lipo -create
"${LIB_BUILD_DIR}/Release-iphoneos/libCITest_Model_Universal.a"
"${LIB_BUILD_DIR}/Release-iphonesimulator/libCITest_Model_Universal.a"
-output "../frameworks/CITest-Model-Universal/Lib/libCITest_Model_Universal.a"
```



Custom iOS framework : motivation

A more native Apple way

Very easy to use and integrate

No source code disclosure

It may contain resources, just like the *.app

Framework is an NSBundle

Flexible versioning and dynamic load (Mac only)

Requires more work to develop and deploy

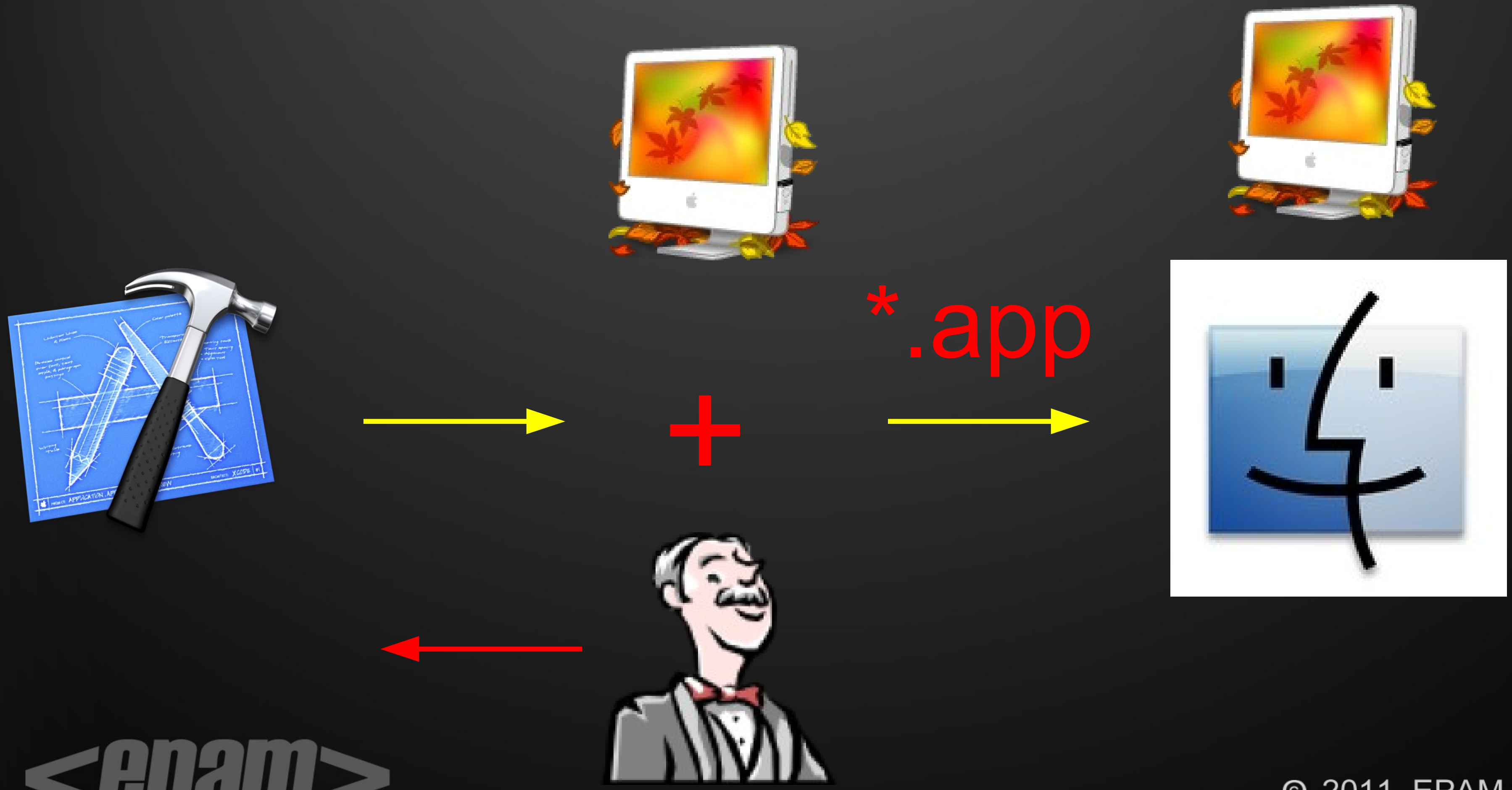
Custom iOS frameworks

1. Yes. You can create and use them
2. For iOS they are linked statically only
3. That's why they have only one version

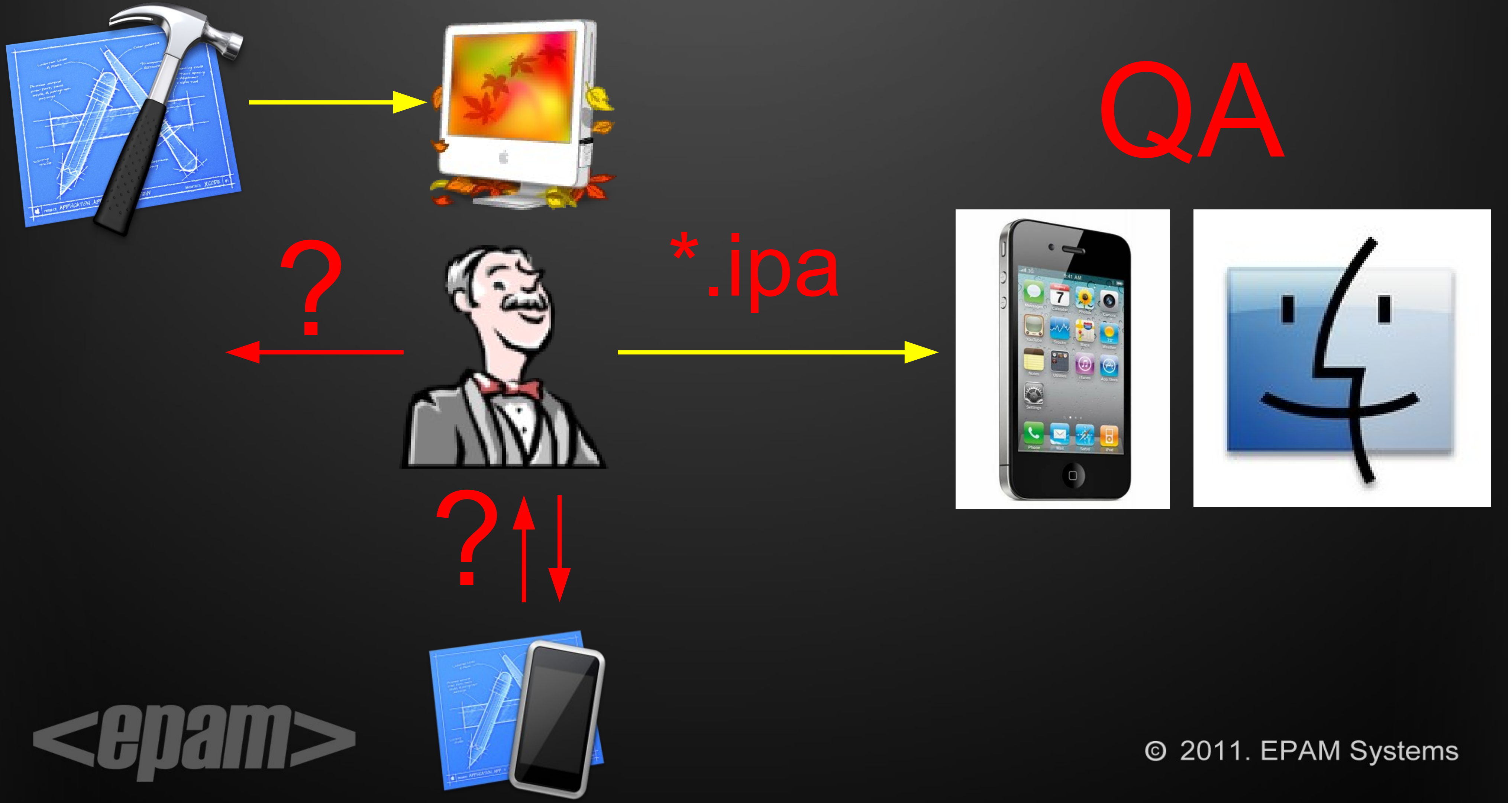
```
MyFramework.framework
|--------> MyFramework  (universal static library)
|-------->Headers        (symlink)
|-------->Resources      (symlink, optional)
|-------->Versions       Actual files should be here
```

Deployment : Desktop vs. Mobile

Desktop Applications



iOS Applications



Inside the *.ipa package

```
CITest.ipa (Zip archive)
|
|---->Payload (a folder)
|         |
|         |  (Contains signature and provisioning)
|         |------>CITest.app
|         |            |
|         |            |------>CITest(.exe)
|         |            |------>*.nib; *.strings; *.plist
|         |            |------>*.jpeg; *.png; ….
```

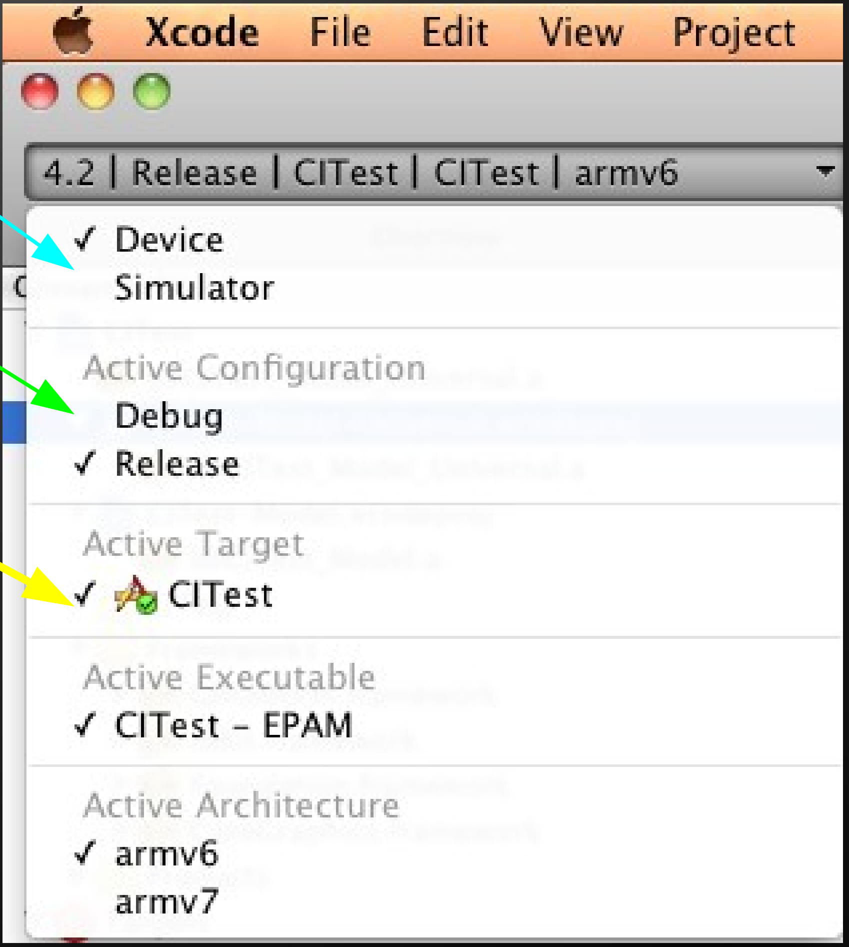
Mobile QA

*.ipa

Testflightapp.com

Building Without xCode GUI

```
xcodebuild -project CITest.xcodeproj
-sdk iphonesimulator4.3
-configuration Release
-target CITest
-parallelizeTargets
clean build
```



Xcode File Edit View Project

4.2 | Release | CITest | CITest | armv6

✓ Device
Simulator

Active Configuration
Debug
✓ Release

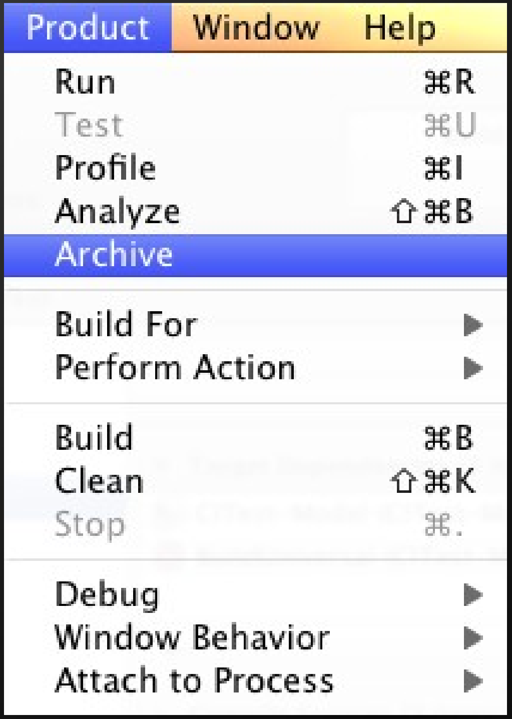
Active Target
✓ CITest

Active Executable
✓ CITest – EPAM

Active Architecture
✓ armv6
armv7

Creating Installable *.ipa File

```
/usr/bin/xcrun -sdk iphoneos PackageApplication
-v "${BUILD_DIR}/Release-iphoneos/CITest.app"
-o "${DEPLOYMENT_DIR}/CITest.ipa"
--sign "${DEVELOPER_NAME}"
--embed "${PROVISONING_PROFILE}"
```



```
DEVELOPER_NAME="iPhone Developer: Oleksandr Dodatko (ABCDEFG123456)"
```

For xCode4 and up

```
/usr/bin/xcrun -sdk iphoneos PackageApplication
-v "${BUILD_DIR}/Release-iphoneos/CITest.app"
-o "${DEPLOYMENT_DIR}/CITest.ipa"
```

~~--sign "${DEVELOPER_NAME}"~~

~~--embed "${PROVISIONING_PROFILE}"~~

How About Unit Testing?

Picking a framework

Running a test

Collecting results

Test Frameworks Chart

	SenTest	Google	GHUnit
Xcode integration	+	+	---
UIKit Support	---	---	+
Bundles support	---	---	+
Xml reports	---	---	+ (lack of support for hudson CI)
Runs on device	+- (Runtime tests only)	+- (Runtime tests only)	+
Runs on simulator	+- (logic tests only)	+- (logic tests only)	+
Debugging (out of box)	---	---	+
UI snapshots comparing	---	+	---

GHUnit Configuration

Add GHUnit.framework

Replace Main.h with the one from GHUnit

Remove “MainNibFile” entry from the info.plits

GHUNIT_AUTORUN

WRITE_JUNIT_XML

GHUNIT_AUTOEXIT

// Not supported in the official GHUNIT

Running a Test

```
iphonesim launch
 "$DEPLOYMENT_DIR/CITest.app"
 4.2
 ipad
```

NOTE : Use only FULL PATH to the app
as shown above

Collecting Test Results

```
TEMP_DIR=$(/usr/bin/getconf DARWIN_USER_TEMP_DIR)
```

All Test results are here :

```
$TEMP_DIR/test-results
```

Terminating the Simulator

```
killall -s -KILL -c "iphonesim"
killall -KILL -c "iphonesim"

killall -s -KILL -c "iPhone Simulator"
killall -KILL -c "iPhone Simulator"
```

Do it before you run a test app

Build and analyze

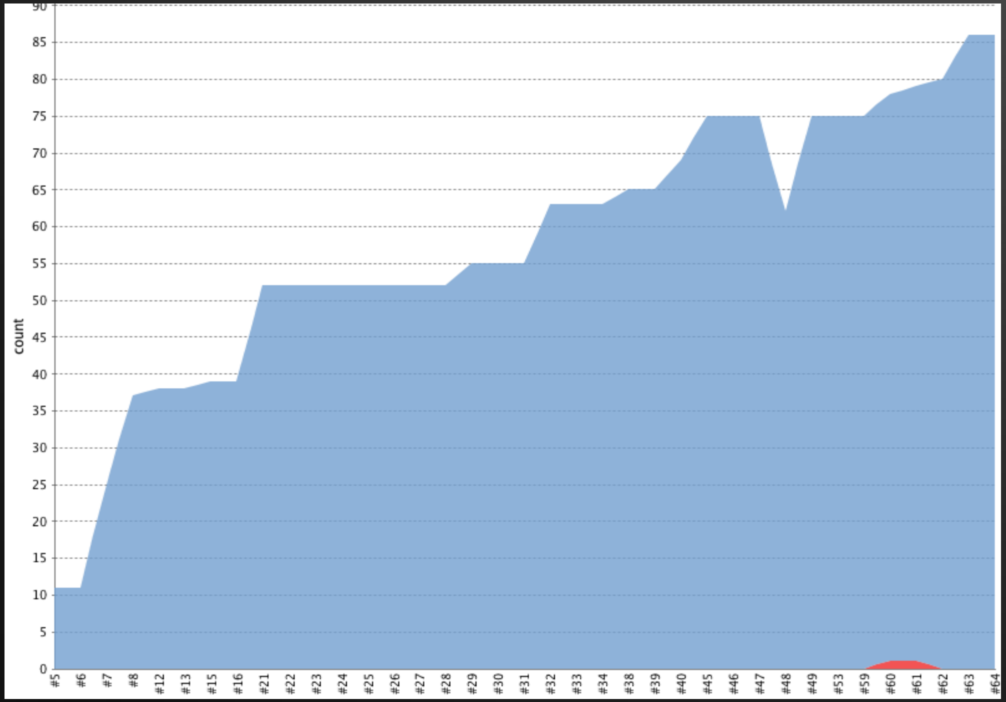
Download and unzip clang

Use scan-build command

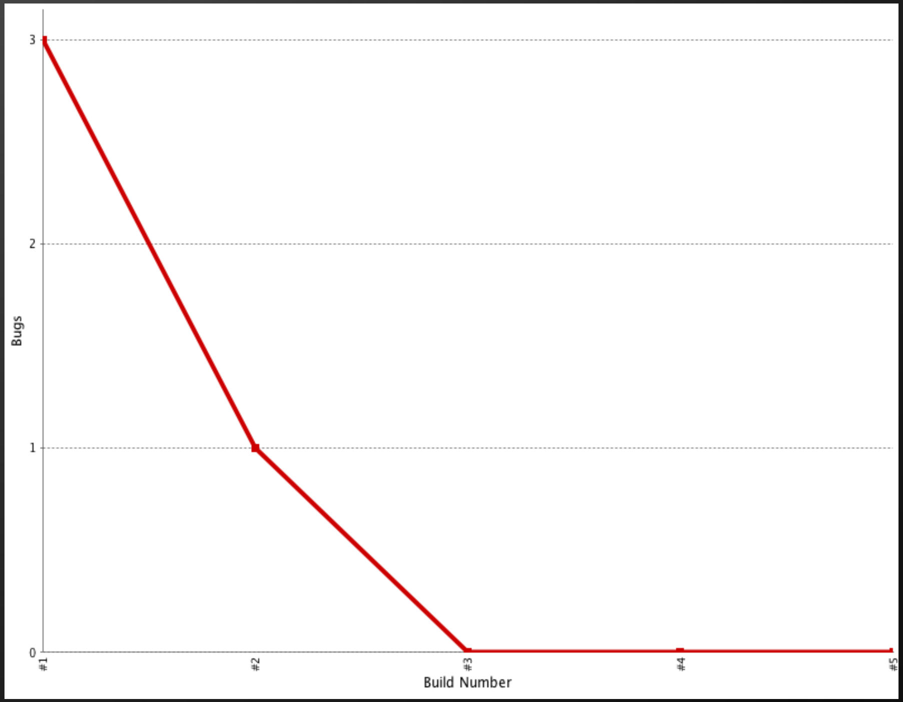
```
scan-build
    -o TargetDir
    xcodebuild <just like for your usual builds>
```

Hudson plug-ins statistics

Unit tests trend

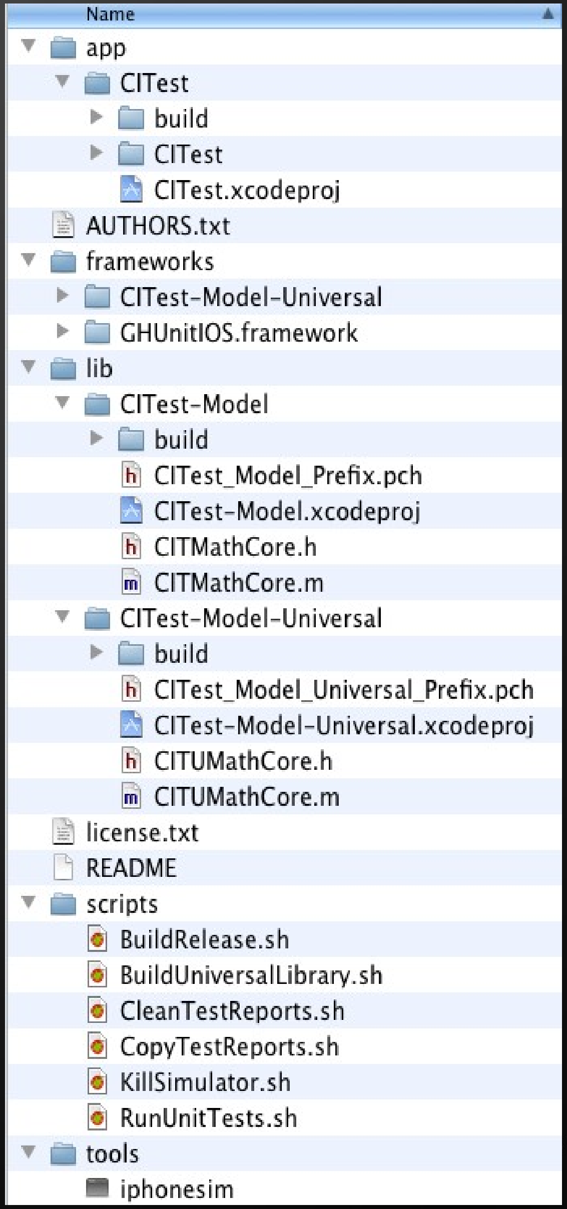


Clang trend



Defining the Project Structure

app
lib
frameworks
scripts
tools
test
certificates
deployment



```
Name
▼ app
    ▼ CITest
        ▶ build
        ▶ CITest
          CITest.xcodeproj
  AUTHORS.txt
▼ frameworks
    ▶ CITest-Model-Universal
    ▶ GHUnitIOS.framework
▼ lib
    ▼ CITest-Model
        ▶ build
          CITest_Model_Prefix.pch
          CITest-Model.xcodeproj
          CITMathCore.h
          CITMathCore.m
    ▼ CITest-Model-Universal
        ▶ build
          CITest_Model_Universal_Prefix.pch
          CITest-Model-Universal.xcodeproj
          CITUMathCore.h
          CITUMathCore.m
  license.txt
  README
▼ scripts
      BuildRelease.sh
      BuildUniversalLibrary.sh
      CleanTestReports.sh
      CopyTestReports.sh
      KillSimulator.sh
      RunUnitTests.sh
▼ tools
      iphonesim
```

Contacts

EPAM systems (Dnipropetrovsk) http://www.epam.com/

Github page : **https://github.com/EmbeddedSources**

https://github.com/EmbeddedSources/iOS-articles

https://github.com/dodikk/iContiniousIntegration

Oleksandr Dodatko

mail/jabber : dodikk88.reg@gmail.com

Skype : alexander.dodatko.work@skype.com

Github page : **https://github.com/dodikk**