

Command Line Xcode

Or: GUI Development Without The G





About Me

- Mobile Lead at MapBox
- 15y CLI / 10y Cocoa / ~3y CLI Xcode
- Mostly work in iOS, though a bit of OS X
- We have a build box for TileMill(.com)

Target Audience

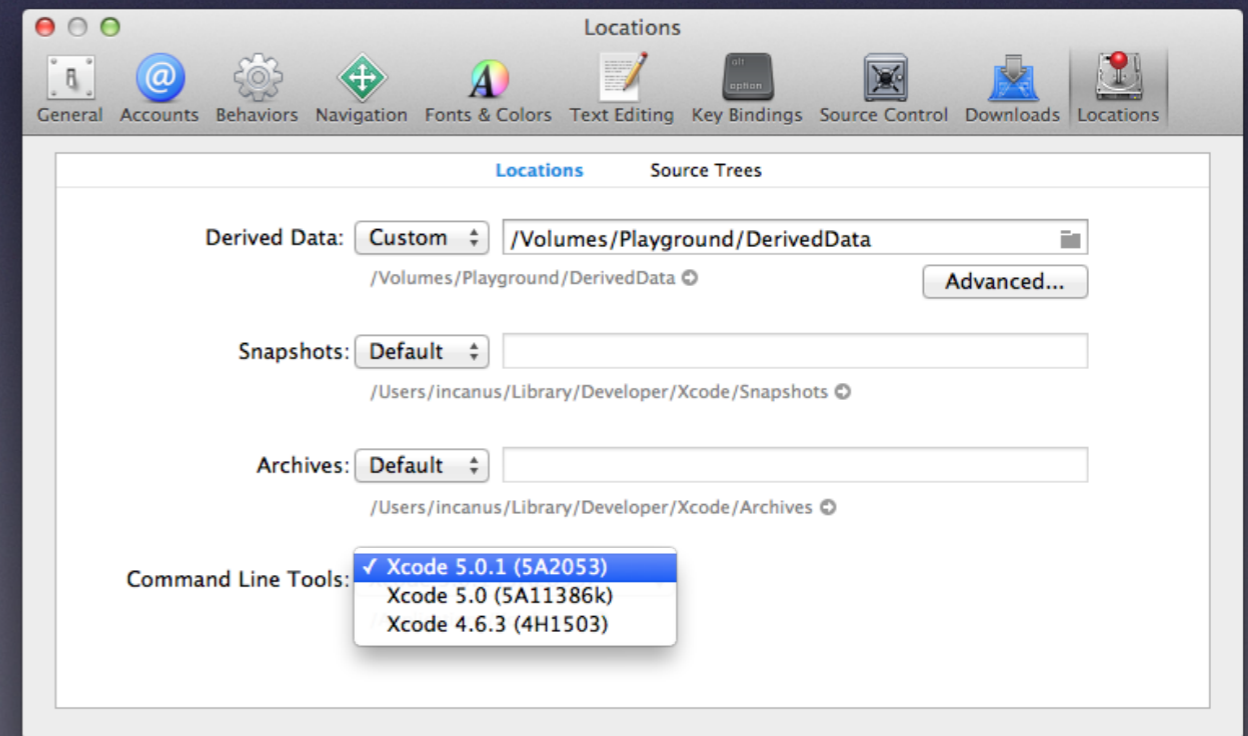
- Cocoa developers
 - You are working on apps in `.xcodproj` files
 - Maybe you want to automate builds or tests
 - Maybe you build frameworks
- CLI tools are enough for straight UNIX stuff

Overview

- Setup & configuration
- Actual building
 - Project files, targets, configurations, architectures, SDKs, etc.
- Assistive tools & extras
 - **lipo**, **hub**, Homebrew, nomad, shell tricks, etc.

Installation

- **Xcode .app** (App Store or **.dmg**)
- Can have multiple installs (e.g. 4.6.3 & 5.0.1)
 - **xcode-select**
 - **DEVELOPER_DIR**
 - Xcode preferences



xcrun

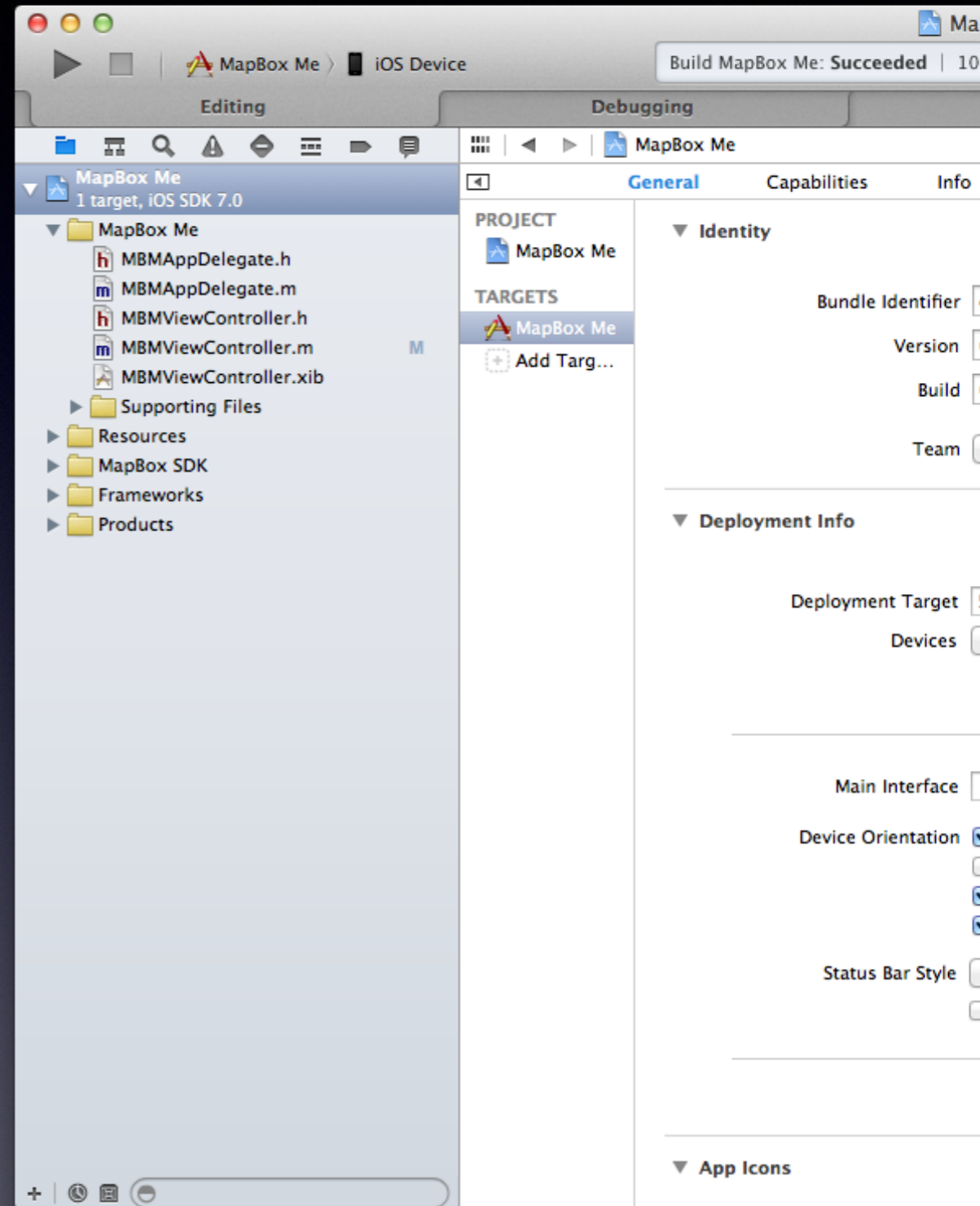
- Very useful for UNIX building
 - e.g. ``xcrun -f clang``
 - Uses a caching system
 - `-n` : Don't use/update cache
 - `-k` : Kill cache

xcodebuild

- The Big Kahuna (that's the technical term)
- You'll spend most of your time in it
- Have a look at the **man** page
 - ``man xcodebuild``

-list

Analogy: having a glance at
the main UI



-project

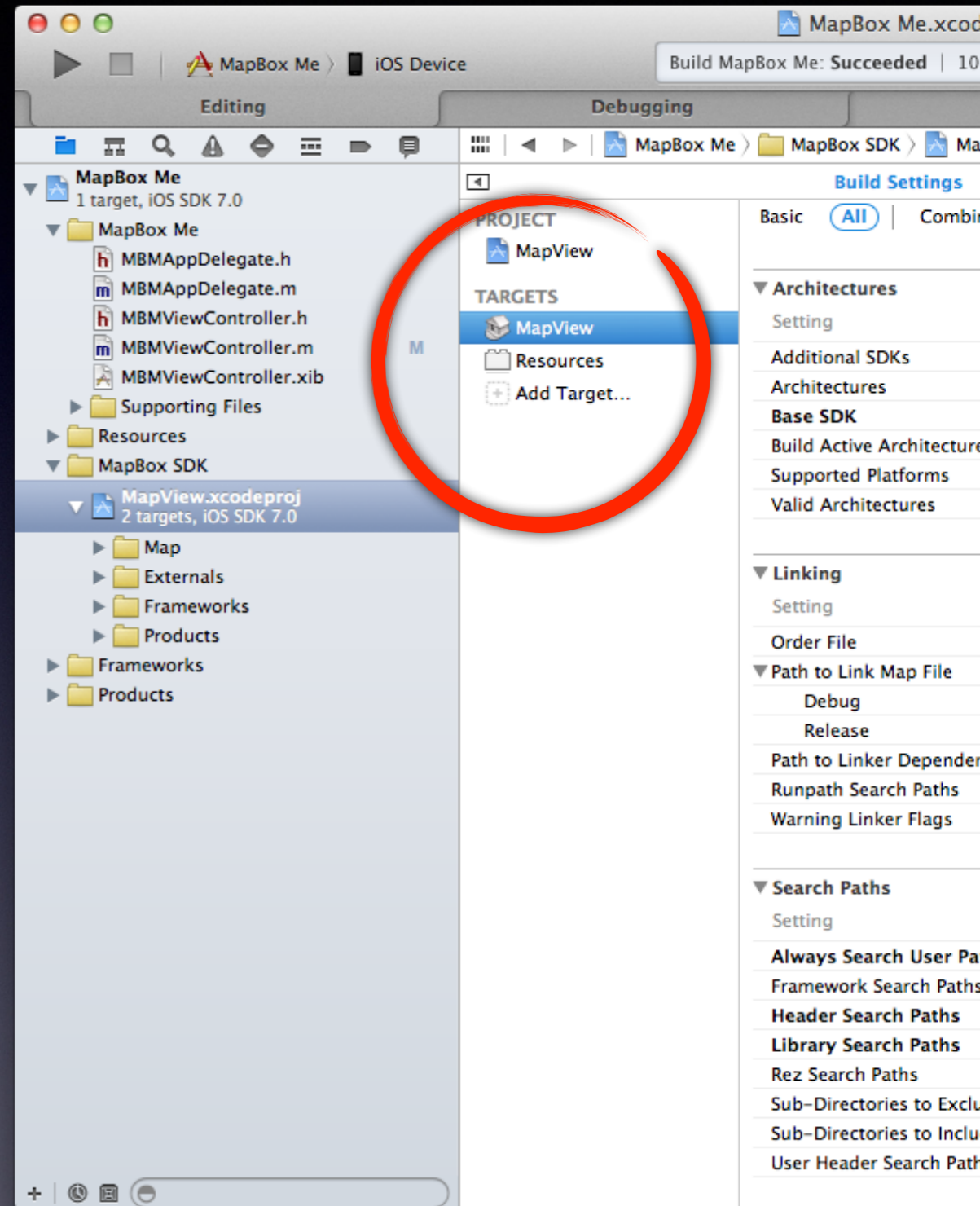
- For when you have multiple project files in the current folder
- Analogy: opening an `.xcodproj`

-workspace

- For when you want to open an `.xcworkspace`
- Personally, I prefer to explicitly specify everything
 - I use workspaces for CocoaPods

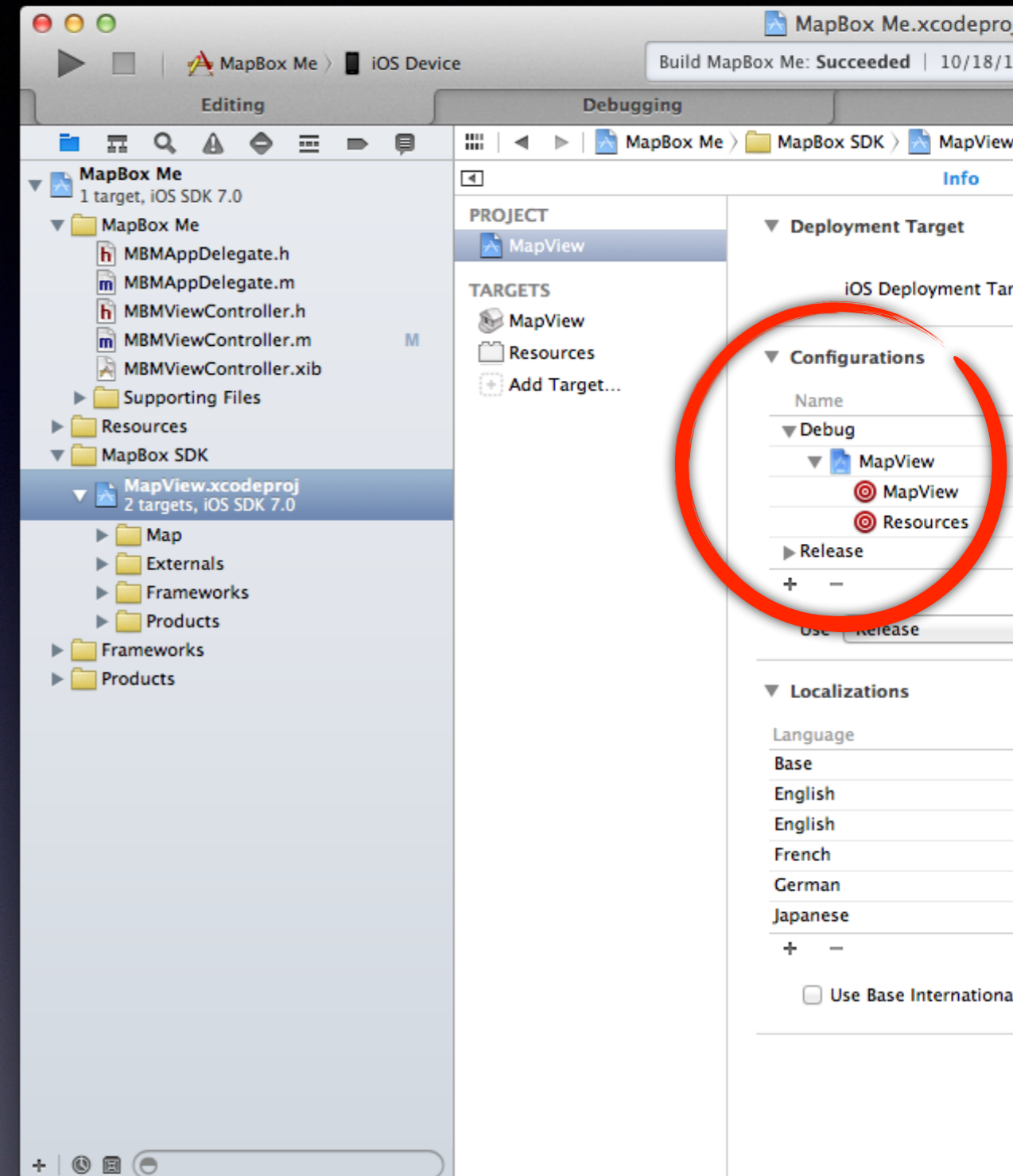
-target

Analogy: target selector
(obviously not rocket
science)



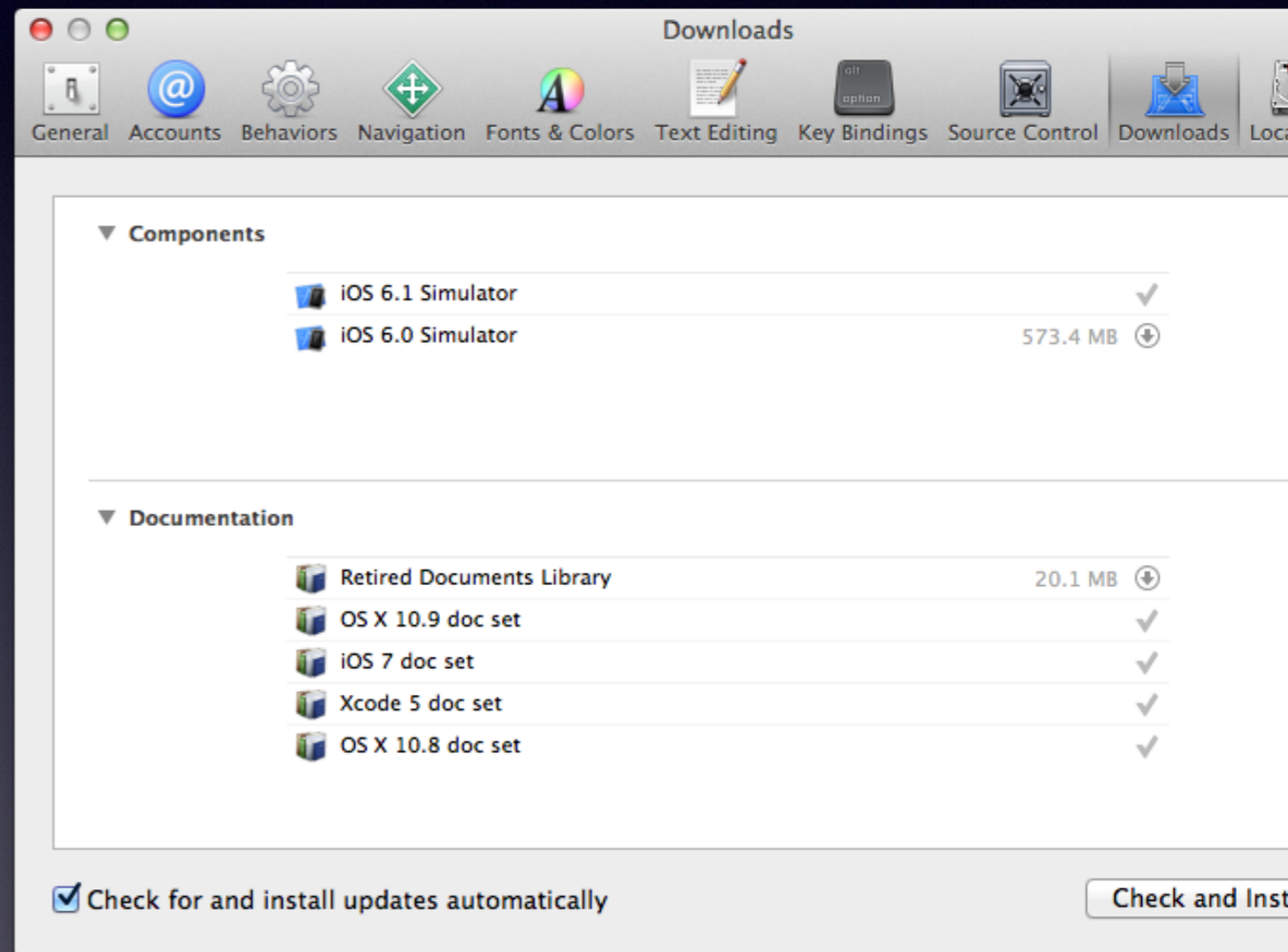
- configuration

Analogy: target configurations
(Release/Debug/Ad Hoc, etc.)



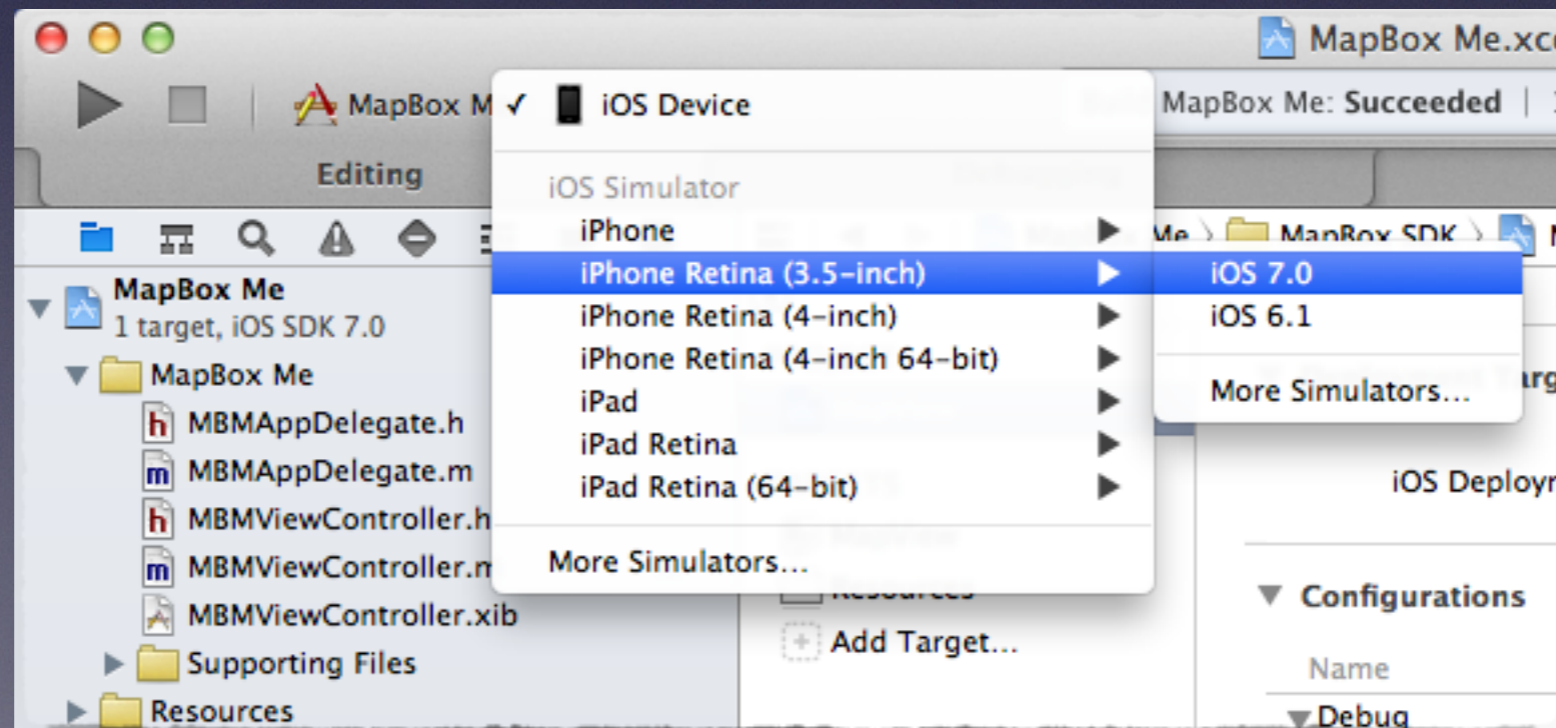
- showsdks

- List installed SDKs
 - `macosx10.6`
 - `iphonios5.0`
 - etc.



- sdk

- iOS 6 or 7, etc.
- `-sdk macosx10.6`
- `-sdk iphoneos5.0`

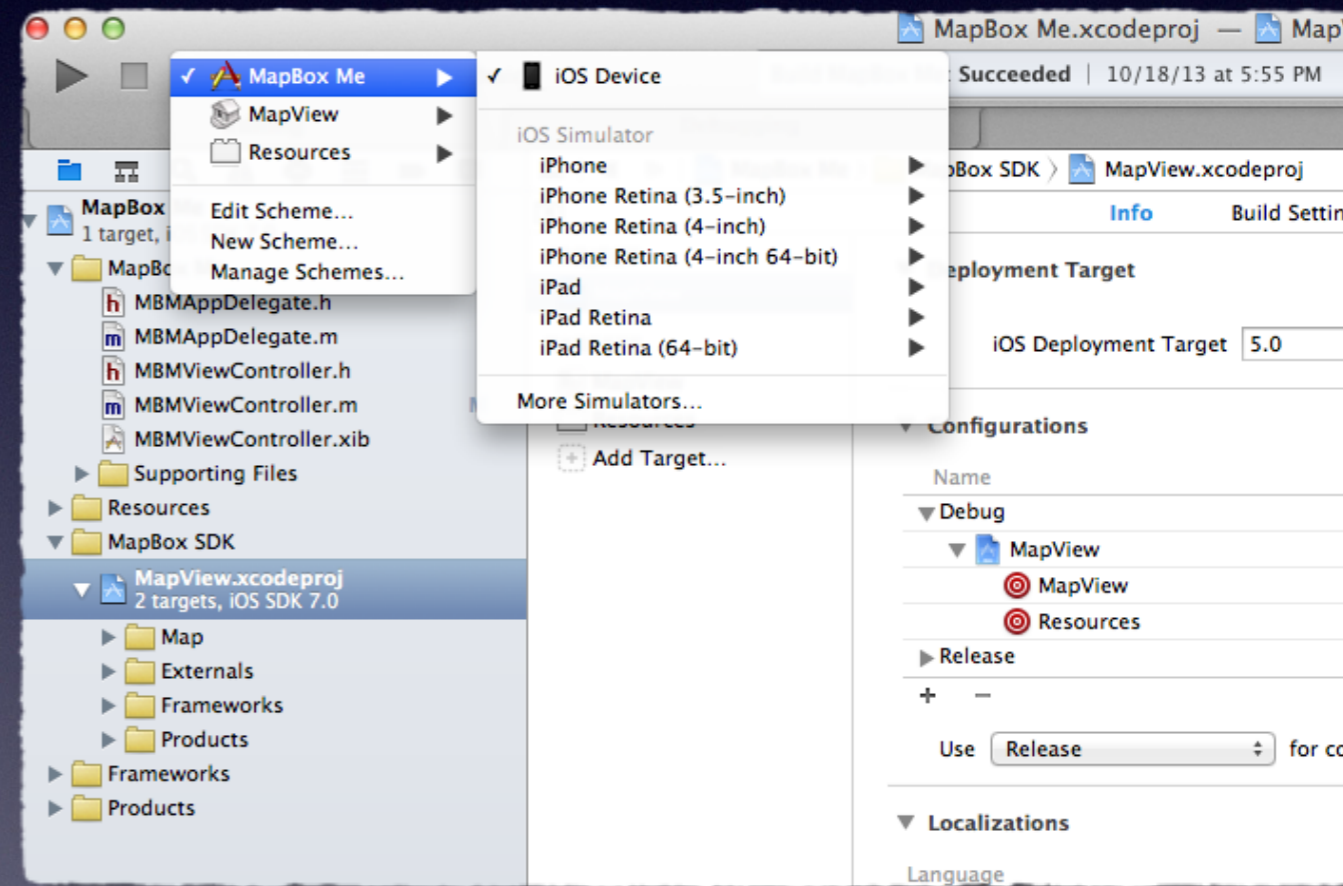


-destination

- Machine architecture
 - `arch=i386`
 - `arch=x86_64` (default)
- Device or simulator
 - `name=Ithildin`
 - `name="iPhone Retina (3.5-inch)"`
- Operating system
 - `OS=7.0`
 - `OS=latest`
- Generic device
 - `name="generic/iOS Device"`
- `-destination-timeout`

- scheme

- Scheme selector
- Obtain from `-list`



-derivedDataPath

- If you don't know what **DerivedData** is, you should probably keep it that way
- Seriously, though: maybe you have a reason to be sure you are working cache-free

-resultBundlePath

- Modify where your built product ends up
- Useful for e.g. plopping it into a shared folder

Archives

- `-exportArchive`
- `-exportFormat`
 - `IPA`, `PKG`, `APP` (guesses intelligently)
- `-archivePath`
- `-exportPath`
- `-exportProvisioningProfile`
- `-exportSigningIdentity`

Build Actions

- `clean`
- `build` (default)
- `analyze`
- `archive`
- `test`
- `installsrc` & `install`

-xcconfig

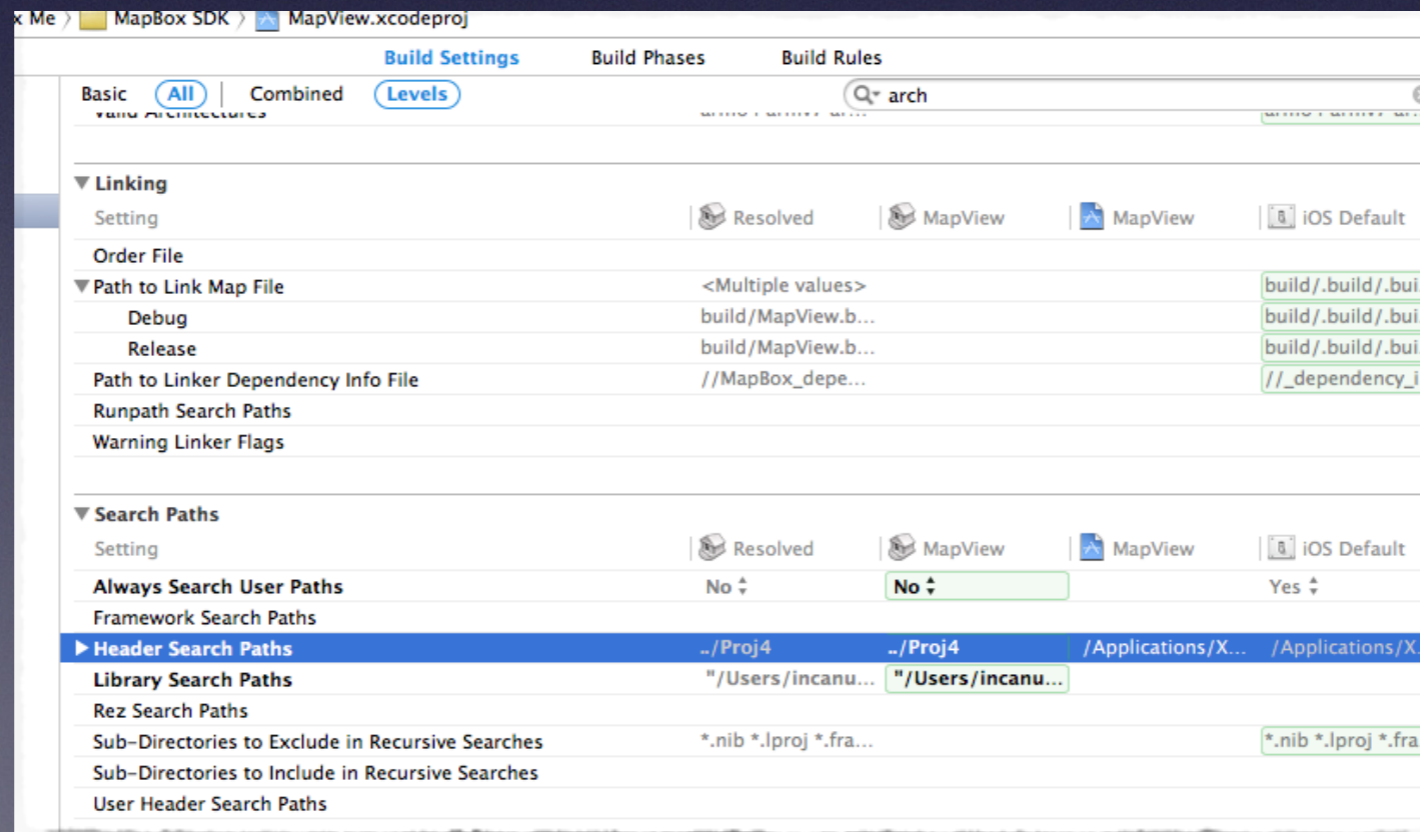
- Useful if you use `.xcconfig` files
- Probably a better route if you use version control and/or have a team

-dry-run & -n

- DON'T TOUCH ANYTHING
- Great way to just try things out
- Thought I can't see what could get hurt

setting=value

- Change build settings
 - e.g. *Header Search Paths*



Extra Goodies

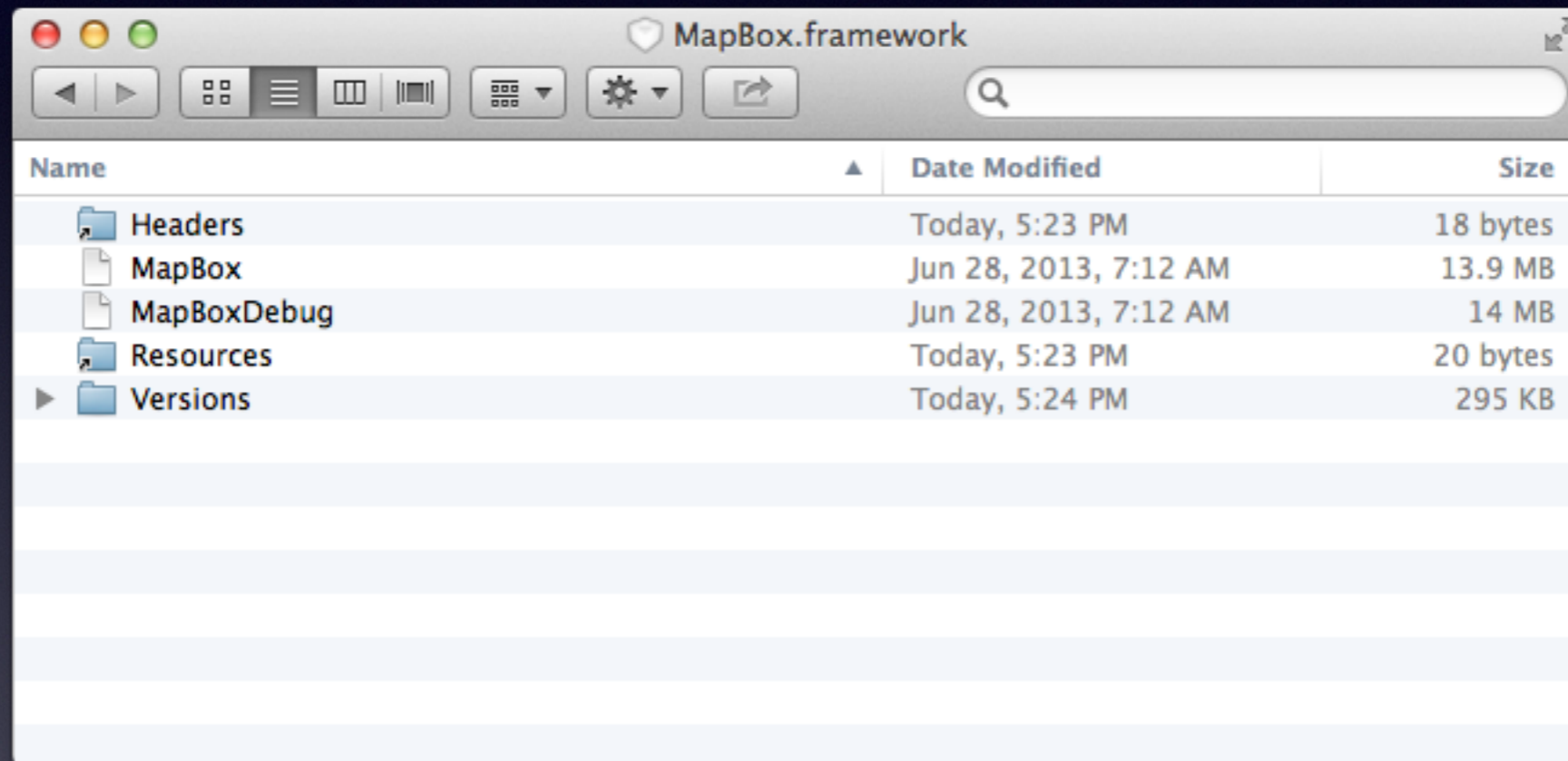
- `lipo`
 - Useful for adding/subtracting slices in fat binaries

lipo Example

```
framework.sh
framework.sh
1 #
2 # build static lib variants
3 #
4 xcodebuild -project MapView/MapView.xcodeproj \
5             -target $TARGET \
6             -configuration Debug \
7             -sdk iphonesimulator${SDK}
8
9             [...] -configuration Debug -sdk iphoneos${SDK}
10            [...] -configuration Release -sdk iphonesimulator${SDK}
11            [...] -configuration Release -sdk iphoneos${SDK}
12
13 #
14 # make fat binaries
15 #
16 lipo -create MapView/build/Debug-iphoneos/lib${FW_NAME}.a \
17         MapView/build/Debug-iphonesimulatr/lib${FW_NAME}.a \
18         -o $FW_FOLDER/${FW_NAME}Debug
19
20 lipo -create MapView/build/Release-iphoneos/lib${FW_NAME}.a \
21         MapView/build/Release-iphonesimulator/lib${FW_NAME}.a \
22         -o $FW_FOLDER/${FW_NAME}
```

Line: 1 | Shell Script (Bash) | Soft Tabs: 4 | No symbols to show for current document.

lipo Example

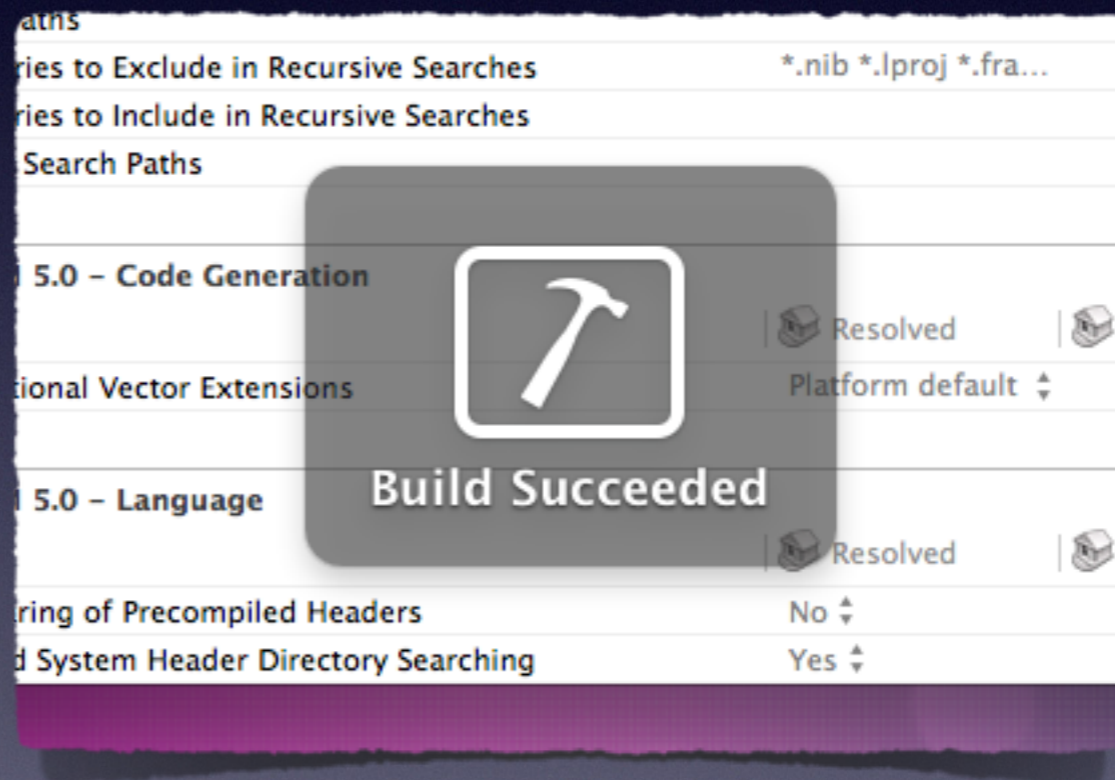


Extra Goodies

- CocoaPods
 - Excellent dependency system
 - But also has a great command-line tool
 - **xcodeproj**
 - Project info, config settings, even resources!

Extra Goodies

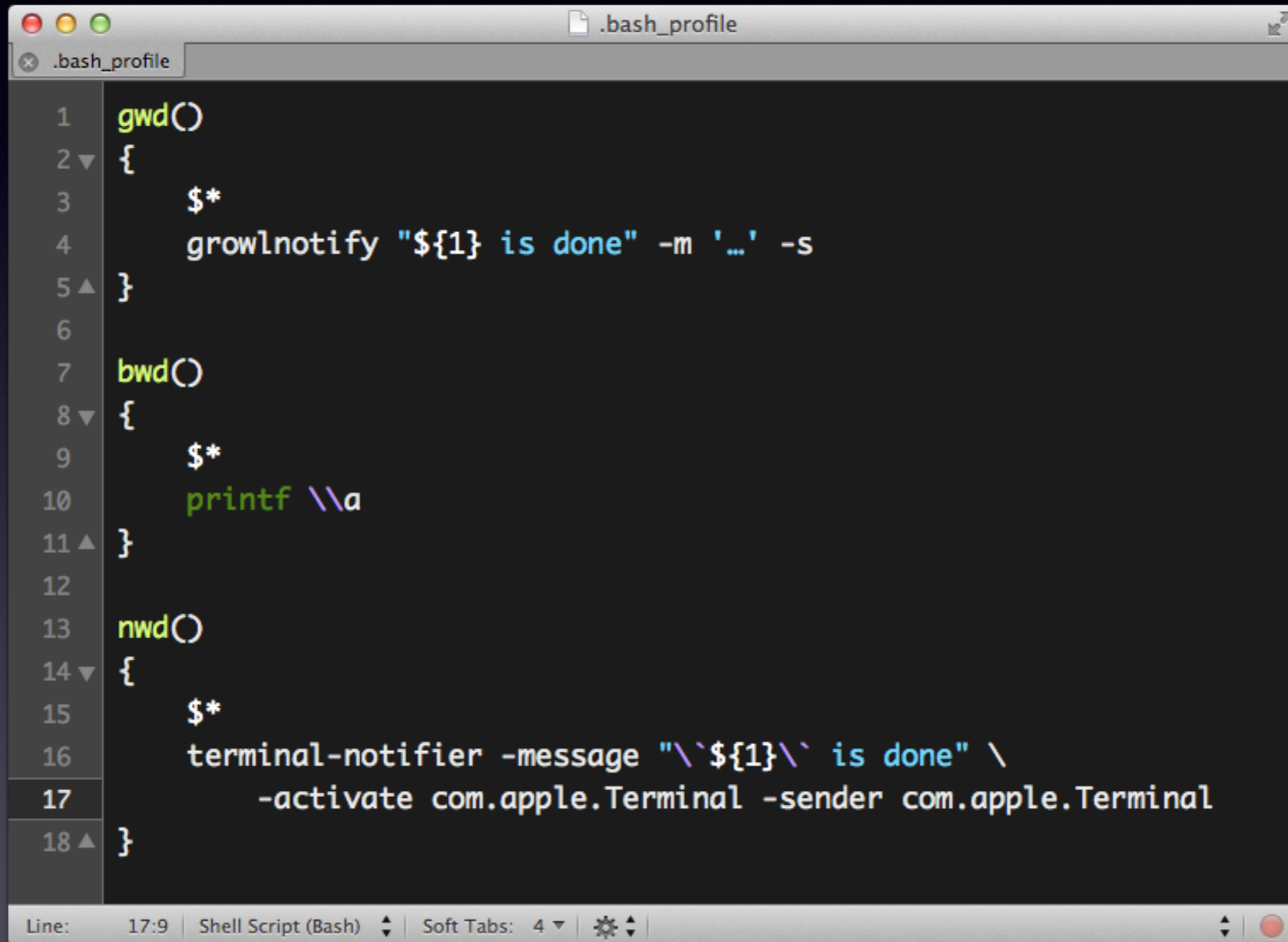
- Notifications
- Bezel
- Notifications



Notifications

- I made some handy shell functions
 - **bwd** - terminal bell
 - **gwd** - Growl when done
 - **nwd** - notify when done
- ``nwd xcodebuild [...]``

Notifications



```
.bash_profile
1  gwd()
2  {
3      $*
4      growlnotify "${1} is done" -m '...' -s
5  }
6
7  bwd()
8  {
9      $*
10     printf \\a
11 }
12
13 nwd()
14 {
15     $*
16     terminal-notifier -message "\\`$1` is done" \
17         -activate com.apple.Terminal -sender com.apple.Terminal
18 }
```

Line: 17:9 | Shell Script (Bash) | Soft Tabs: 4

Notifications

The screenshot shows the GitHub repository page for `alloy/terminal-notifier`. The page title is "Send User Notifications on Mac OS X 10.8 from the command-line." The repository statistics show 105 commits, 1 branch, 9 releases, and 13 contributors. The current branch is `master`. The file list includes `Ruby`, `Terminal Notifier.xcodeproj`, `Terminal Notifier`, `.gitignore`, `README.markdown`, and `Terminal.icns`. The `README.markdown` file is selected, showing the repository name `terminal-notifier` and a description: "terminal-notifier is a command-line tool to send Mac OS X User Notifications, which are available in Mac OS X 10.8 and higher." The right sidebar contains navigation links for Code, Issues (4), Pull Requests (1), Wiki, Pulse, Graphs, and Network. The SSH clone URL is `git@github.com:alloy/terminal-notifier`. There are buttons for "Clone in Desktop" and "Download ZIP".

alloy / terminal-notifier

105 commits 1 branch 9 releases 13 contributors

branch: master terminal-notifier

Add homebrew installation to README

latest commit 2999eede0b

Ruby	Bump to 1.5.1	2 months ago
Terminal Notifier.xcodeproj	Emoji characters in scheme command-line args crash Xcode (4 & 5) on 1...	2 months ago
Terminal Notifier	Bump to 1.5.1	2 months ago
.gitignore	Add Ruby bindings	a year ago
README.markdown	Add homebrew installation to README	9 days ago
Terminal.icns	Use Apple's Terminal.app icon. Not shown in notification yet, might b...	a year ago

README.markdown

terminal-notifier

terminal-notifier is a command-line tool to send Mac OS X User Notifications, which are available in Mac OS X 10.8 and higher.

Code

- Issues 4
- Pull Requests 1
- Wiki

Pulse

Graphs

Network

SSH clone URL

git@github.com:alloy/terminal-notifier

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP

Extra Goodies

- **hub**
 - **create** - create repo on GitHub & set origin
 - **browse** - open GitHub page in browser
 - **compare a...b** - open GitHub compare page
 - **pull-request & fork** - YMMV

Extra Goodies

- Xcode ships with nice Git Bash completion
- Add to `~/ .bash_profile`:
 - `. `xcode-select -p` /usr/share/git-core/git-completion.bash`
 - `. `xcode-select -p` /usr/share/git-core/git-prompt.sh`

Extra Goodies

- nomad - <http://nomad-cli.com>
- `gem install nomad-cli`



CUPERTINO

Automate administrative tasks that you would normally have to do through the Apple Dev Center website. Life's too short to manage device identifiers by hand!

```
$ ios devices:list
```




HOUSTON

Send push notifications from the command line. Simply provide your credentials, construct your message, and 3...2...1... blastoff.

```
$ apn push "<token>" -c /path/to/cert.pem -m "Hello!"
```




DUBAI

Generate Passbook .pkpass files with ease. Rapidly iterate on the design and content of your passes, or generate one-offs on the fly.

```
$ pk generate Example.pass -T boarding-pass
```




VENICE

Secure your In-App-Purchases by verifying App Store purchase receipts, and retrieving the information associated with receipt data.

```
$ iap verify /path/to/receipt
```




SHENZHEN

Create development builds and then distribute their .ipa files over TestFlight, HockeyApp, Amazon S3, or FTP. Get new builds out to testers and enterprises in seconds.

```
$ ipa build && ipa distribute
```


Conclusion

- Xcode is very scriptable
- Apple has put a lot of time into this
 - Hint: someone uses it internally
- It can actually be fun!

Thanks!

- @incanus77
- justin@mapbox.com
- I'll post these slides online