Building from the Command Line with Xcode

Session 404

Anders Bertelrud
Xcode Architect

Roadmap

- Why use the command line?

Roadmap

- Why use the command line?
- Using `xcodebuild`

Roadmap

- Why use the command line?
- Using `xcodebuild`
- Customizing your project

Roadmap

- Why use the command line?
- Using `xcodebuild`
- Customizing your project
- Automating `xcodebuild`

Roadmap

- Why use the command line?
- Using `xcodebuild`
- Customizing your project
- Automating `xcodebuild`
- Working outside Xcode

Why Use the Command Line?

Why Use the Command Line?

Why Use the Command Line?

- Automation
  - Nightly release builds
  - Continuous Integration systems

Why Use the Command Line?

- Automation
  - Nightly release builds
  - Continuous Integration systems
- Make-based open source development
  - Cross-platform code
  - MacPorts, Fink, Homebrew…

Using `xcodebuild`

What Is `xcodebuild`?

What Is `xcodebuild`?

- Command line access to Xcode IDE

What Is `xcodebuild`?

- Command line access to Xcode IDE
- Works with projects and workspaces

What Is `xcodebuild`?

- Command line access to Xcode IDE
- Works with projects and workspaces
- Batch operations
  - Build
  - Archive
  - Query

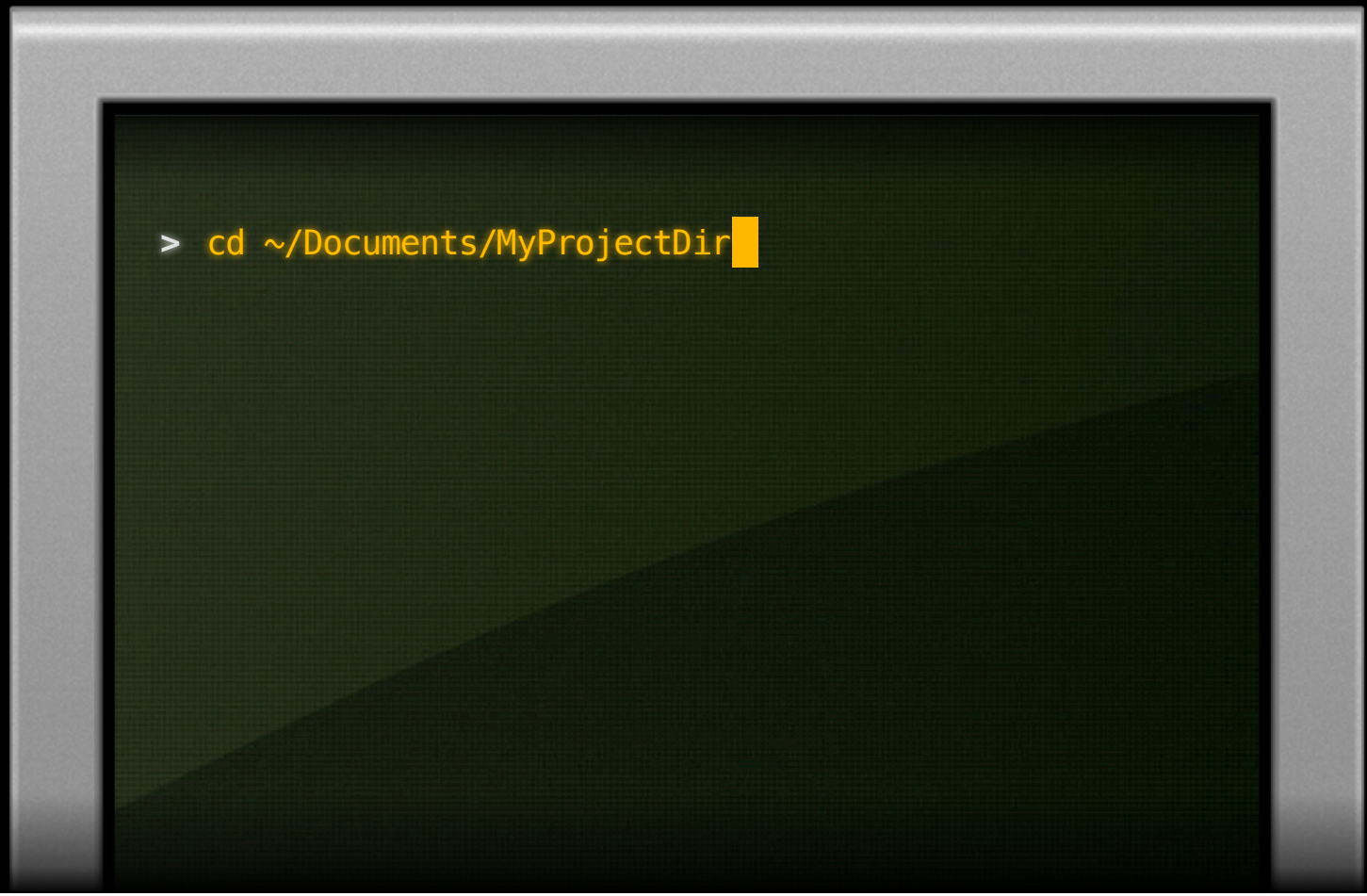
What Is `xcodebuild`?

- Command line access to Xcode IDE
- Works with projects and workspaces
- Batch operations
  - Build
  - Archive
  - Query
- Use `man xcodebuild` for the details

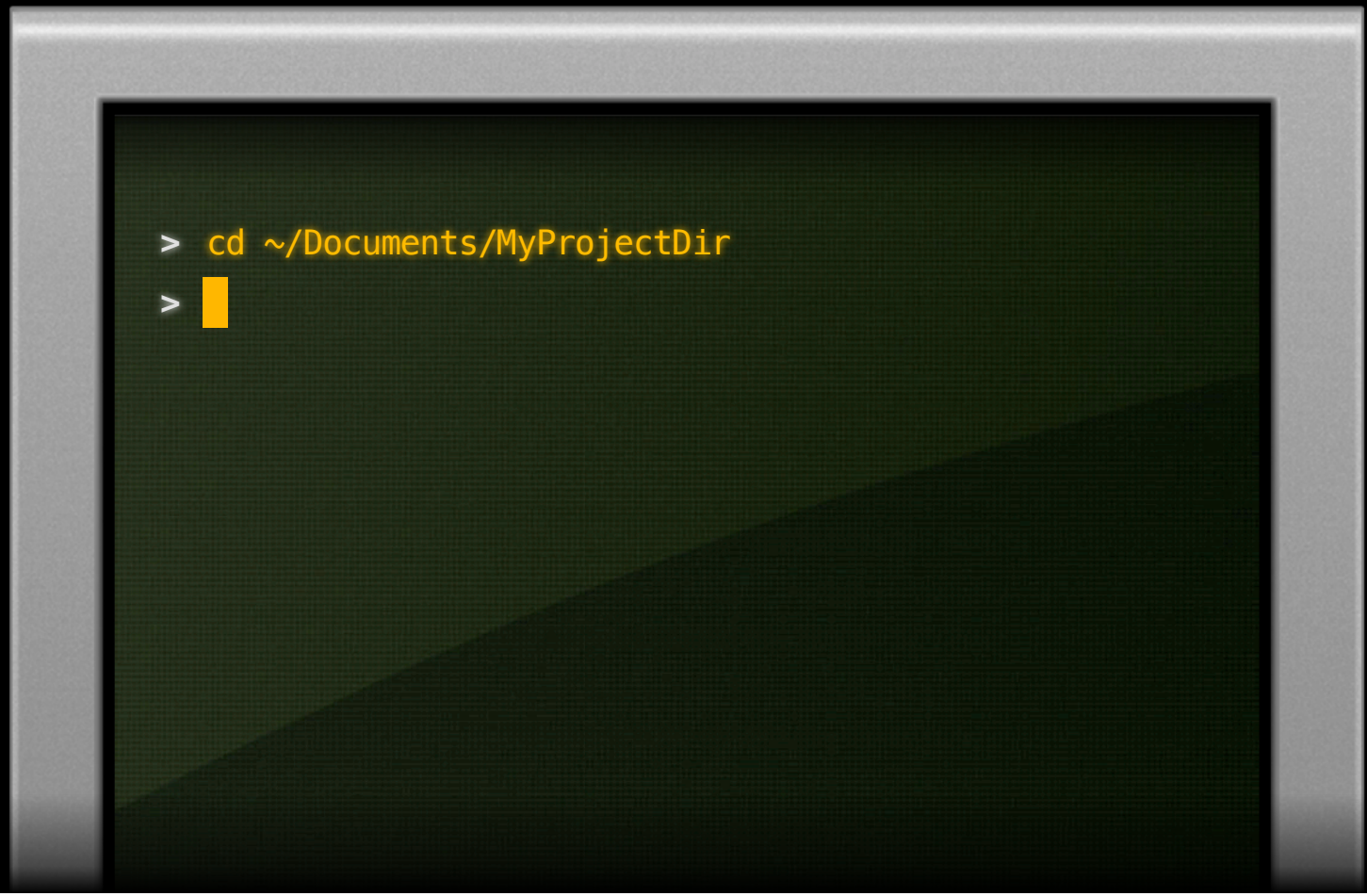
Using `xcodebuild`

Using `xcodebuild`

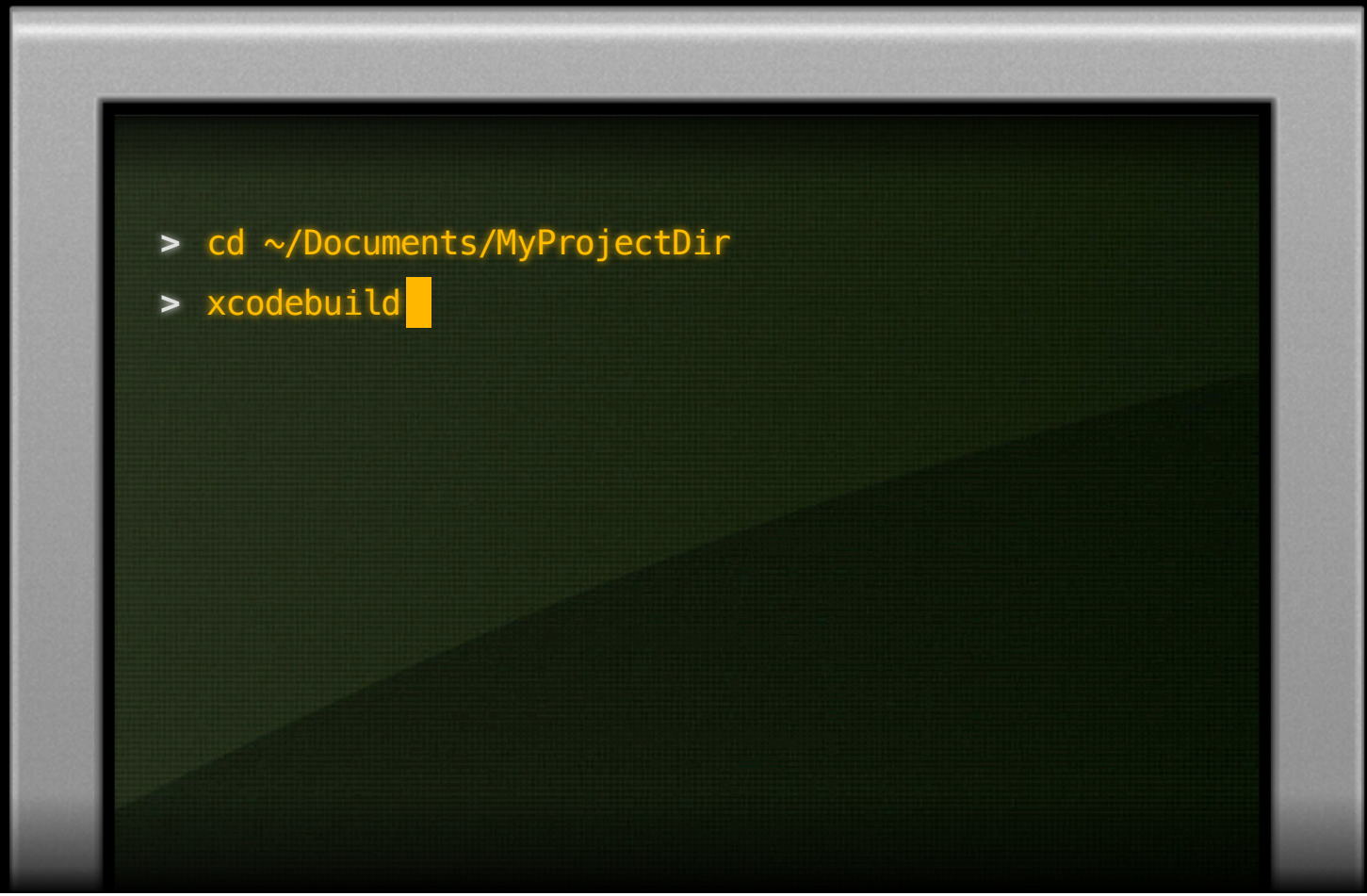
Using `xcodebuild`



Using `xcodebuild`



Using `xcodebuild`



Using `xcodebuild`

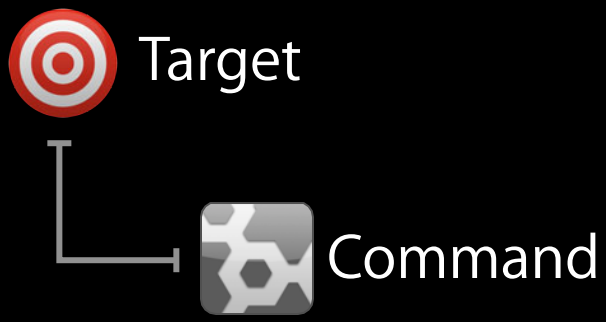
```
> cd ~/Documents/MyProjectDir
> xcodebuild
=== BUILD NATIVE TARGET Baffle OSX OF PROJECT Baffle
OSX WITH CONFIGURATION Debug ===

ProcessPCH build/Intermediates/PrecompiledHeaders/Baff
leCocoa_Prefix-dkhujustkymhfqbdihrzilncxtxa/BaffleCoco
a_Prefix.pch.pth OtherSources/BaffleCocoa_Prefix.pch n
ormal x86_64 objective-c com.apple.compilers.llvm.clan
g.1_0.compiler

                          :
```

Build Log Structure

Build Log Structure

Target

Build Log Structure

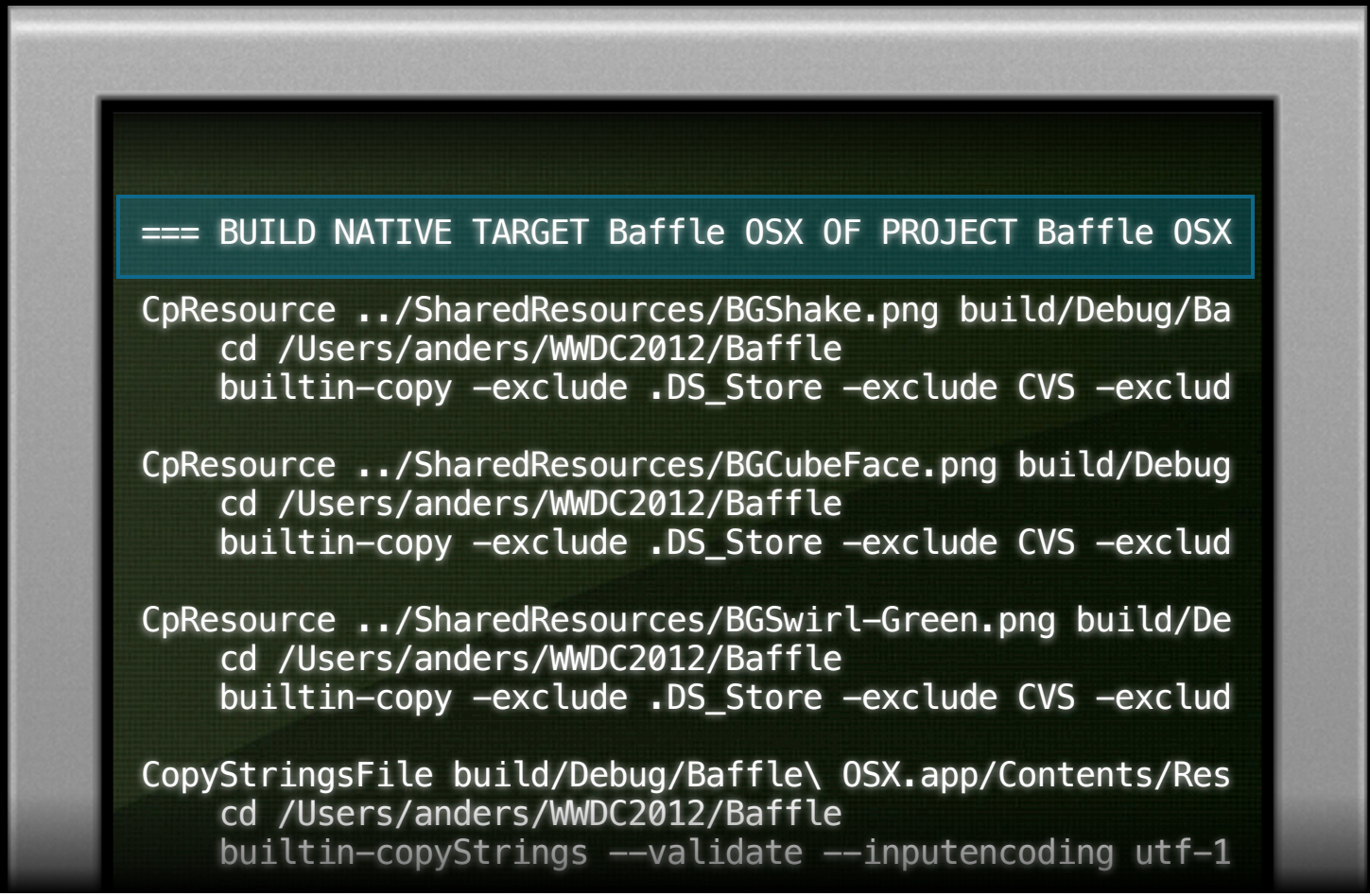


Build Log Structure

Build Logs in the Console

```
=== BUILD NATIVE TARGET Baffle OSX OF PROJECT Baffle OSX

CpResource ../SharedResources/BGShake.png build/Debug/Ba
    cd /Users/anders/WWDC2012/Baffle
    builtin-copy -exclude .DS_Store -exclude CVS -exclud

CpResource ../SharedResources/BGCubeFace.png build/Debug
    cd /Users/anders/WWDC2012/Baffle
    builtin-copy -exclude .DS_Store -exclude CVS -exclud

CpResource ../SharedResources/BGSwirl-Green.png build/De
    cd /Users/anders/WWDC2012/Baffle
    builtin-copy -exclude .DS_Store -exclude CVS -exclud

CopyStringsFile build/Debug/Baffle\ OSX.app/Contents/Res
    cd /Users/anders/WWDC2012/Baffle
    builtin-copyStrings --validate --inputencoding utf-1
```

Build Logs in the Console



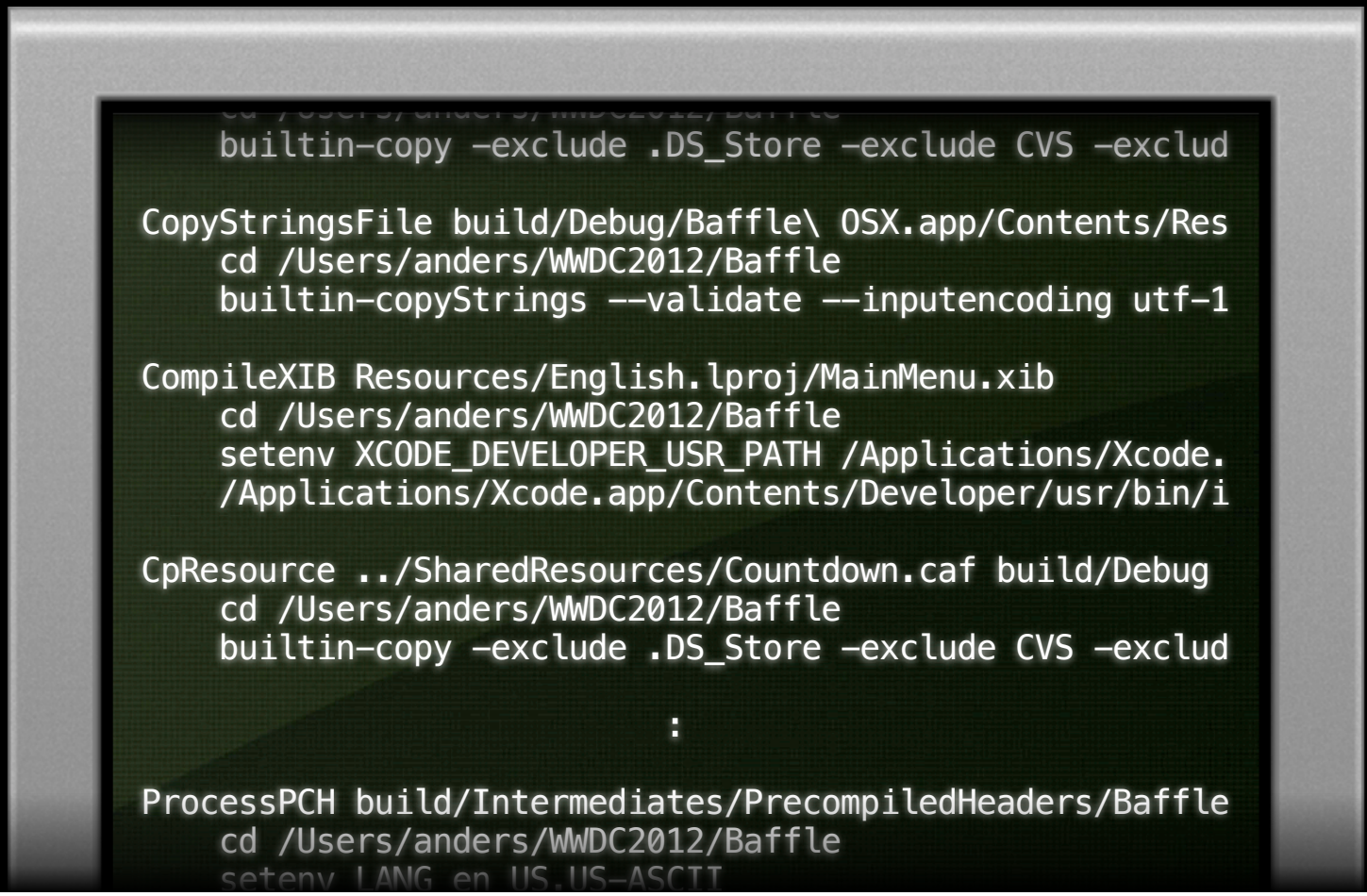
Build Logs in the Console

```
=== BUILD NATIVE TARGET Baffle OSX OF PROJECT Baffle OSX

CpResource ../SharedResources/BGShake.png build/Debug/Ba
    cd /Users/anders/WWDC2012/Baffle
    builtin-copy -exclude .DS_Store -exclude CVS -exclud

CpResource ../SharedResources/BGCubeFace.png build/Debug
    cd /Users/anders/WWDC2012/Baffle
    builtin-copy -exclude .DS_Store -exclude CVS -exclud

CpResource ../SharedResources/BGSwirl-Green.png build/De
    cd /Users/anders/WWDC2012/Baffle
    builtin-copy -exclude .DS_Store -exclude CVS -exclud

CopyStringsFile build/Debug/Baffle\ OSX.app/Contents/Res
    cd /Users/anders/WWDC2012/Baffle
    builtin-copyStrings --validate --inputencoding utf-1
```

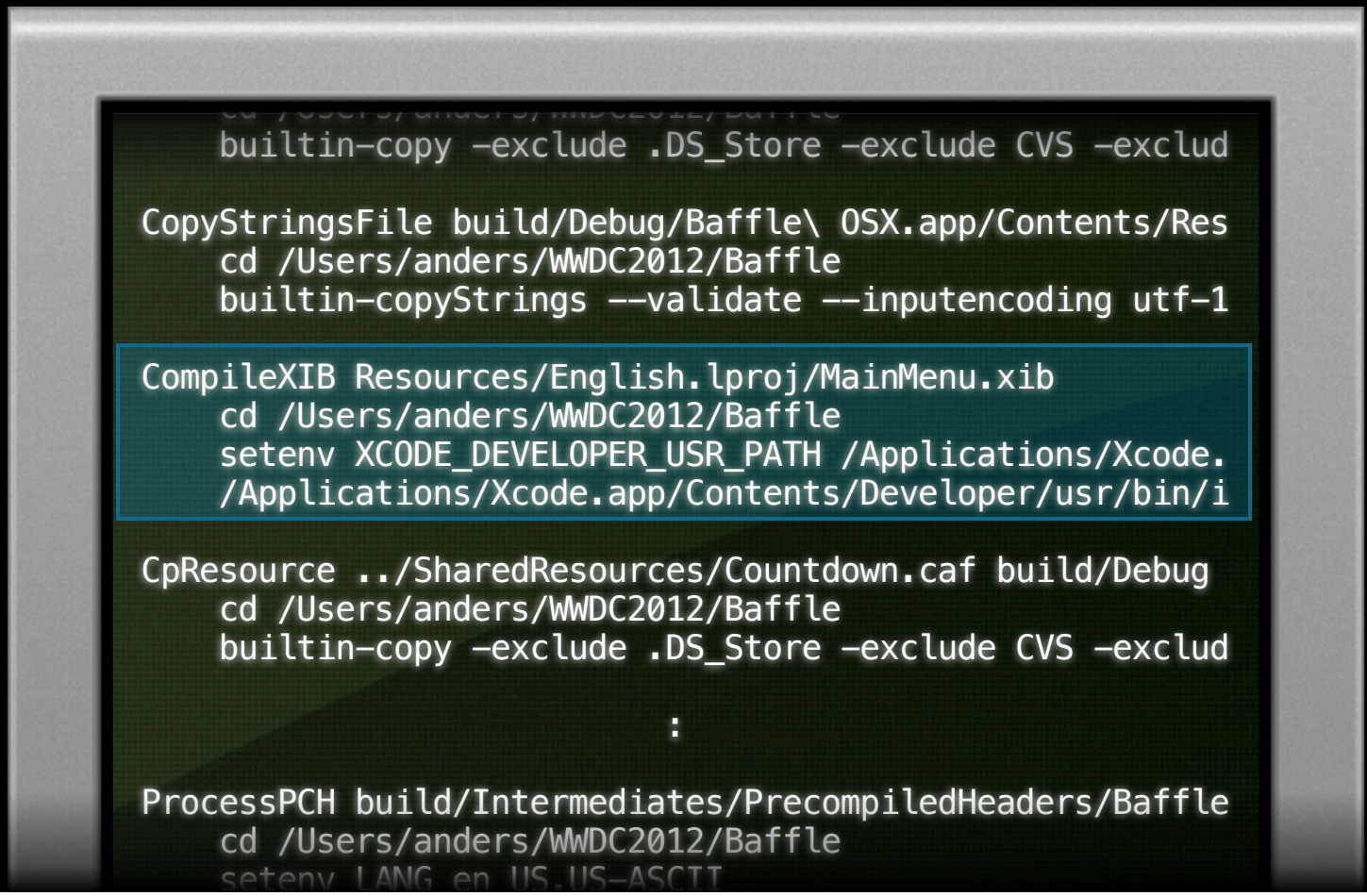
Build Logs in the Console

```
=== BUILD NATIVE TARGET Baffle OSX OF PROJECT Baffle OSX

CpResource ../SharedResources/BGShake.png build/Debug/Ba
    cd /Users/anders/WWDC2012/Baffle
    builtin-copy -exclude .DS_Store -exclude CVS -exclud

CpResource ../SharedResources/BGCubeFace.png build/Debug
    cd /Users/anders/WWDC2012/Baffle
    builtin-copy -exclude .DS_Store -exclude CVS -exclud

CpResource ../SharedResources/BGSwirl-Green.png build/De
    cd /Users/anders/WWDC2012/Baffle
    builtin-copy -exclude .DS_Store -exclude CVS -exclud

CopyStringsFile build/Debug/Baffle\ OSX.app/Contents/Res
    cd /Users/anders/WWDC2012/Baffle
    builtin-copyStrings --validate --inputencoding utf-1
```

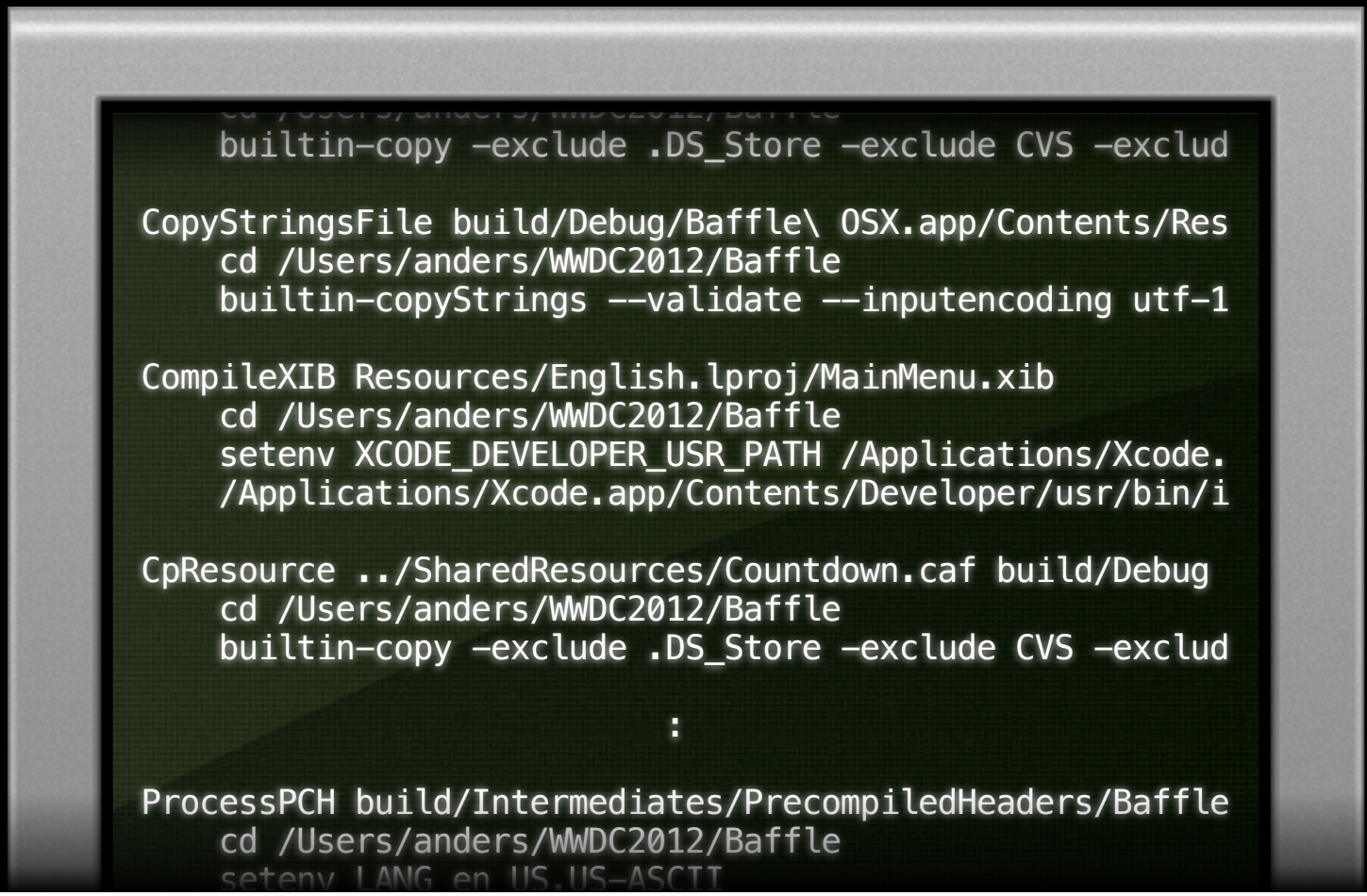
Build Logs in the Console



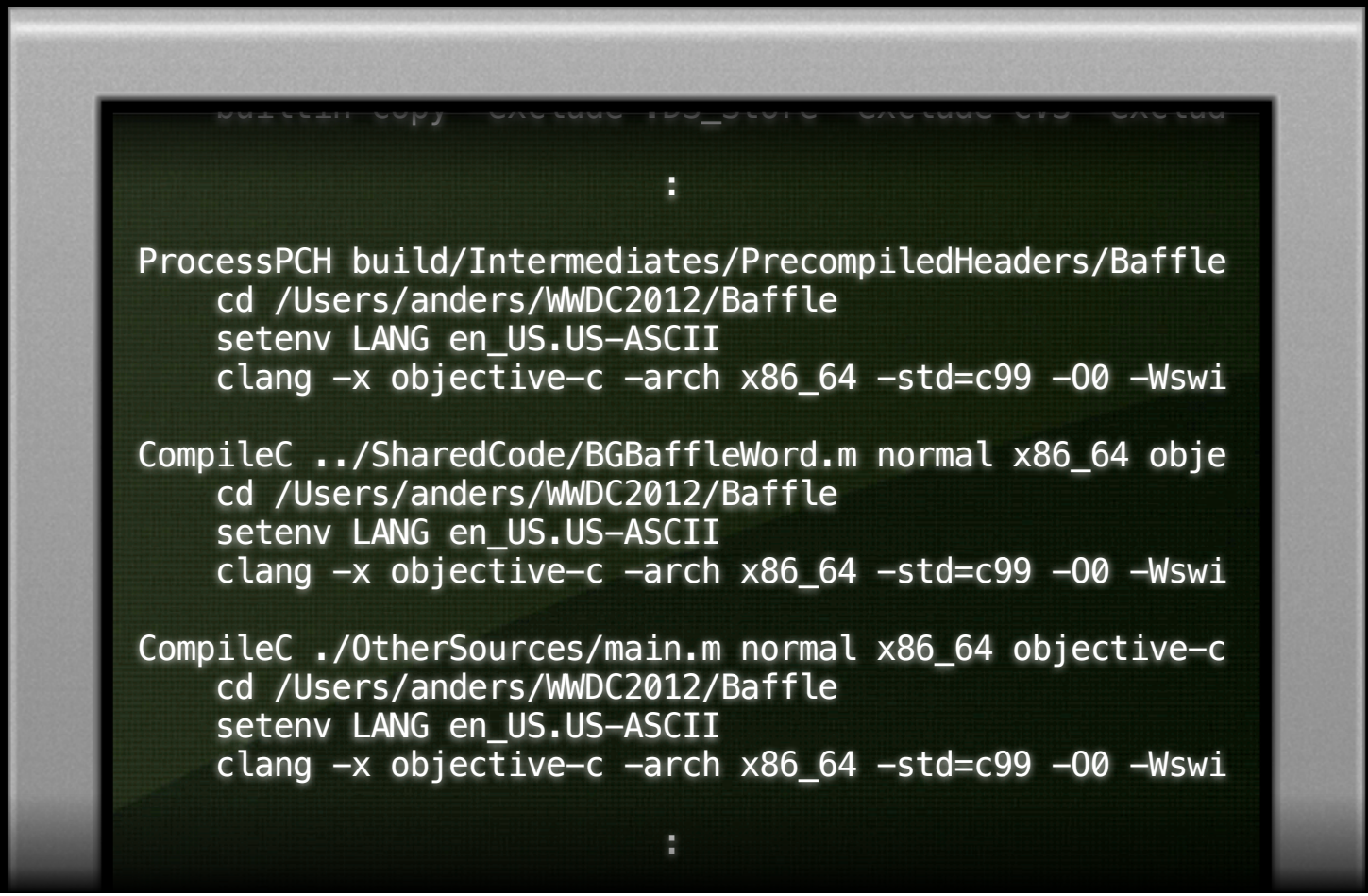
Build Logs in the Console



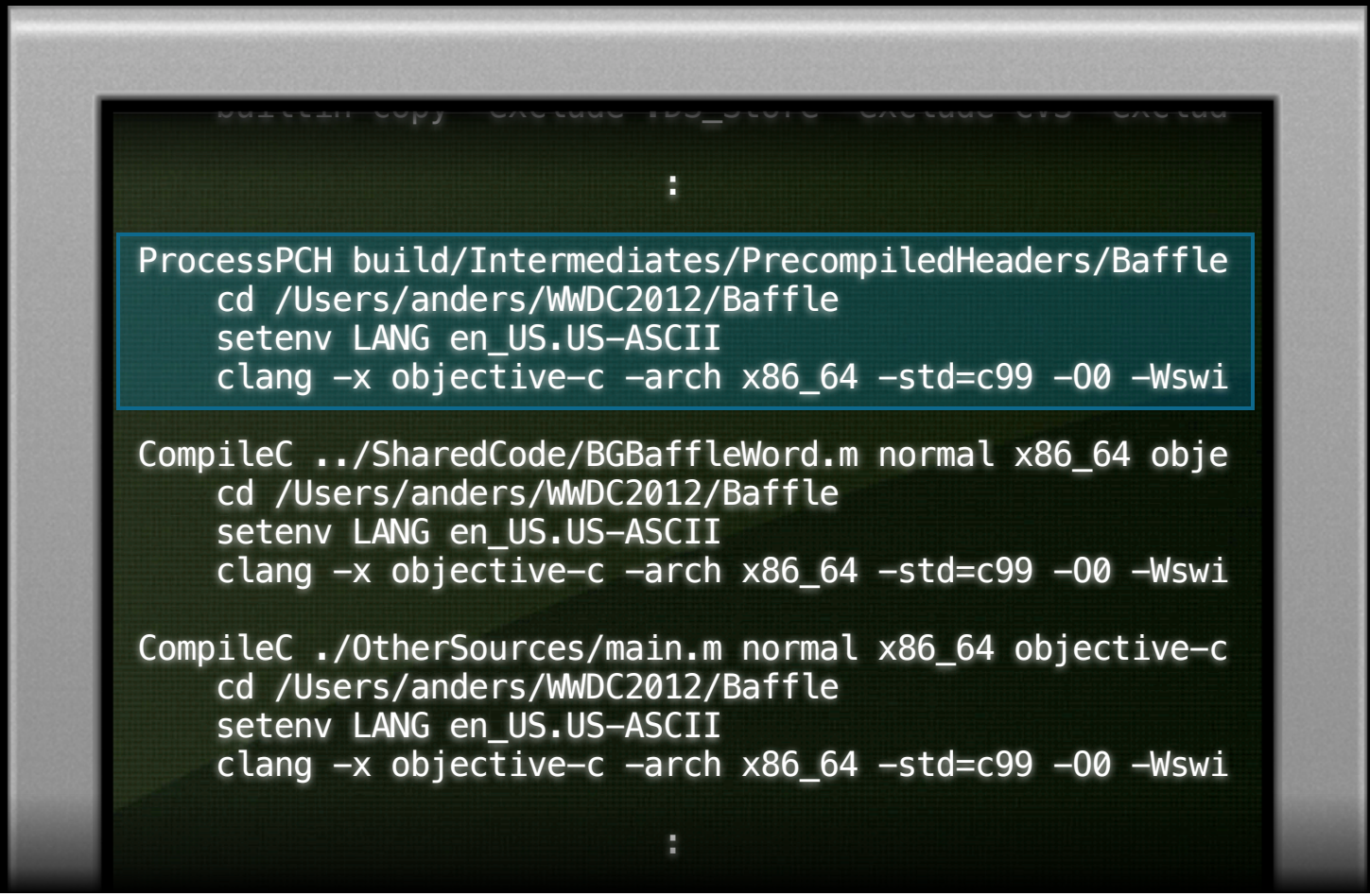
Build Logs in the Console



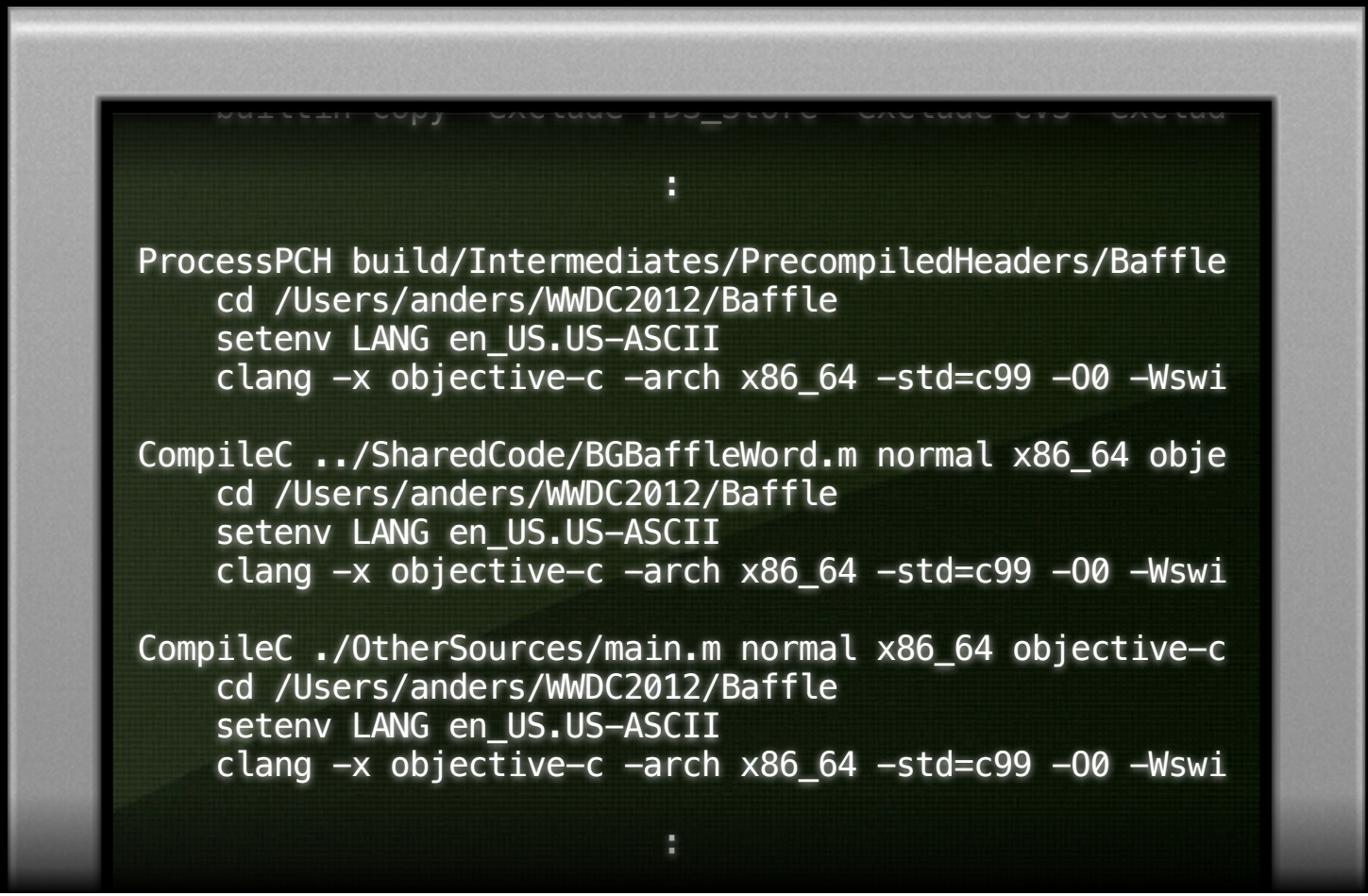
Build Logs in the Console



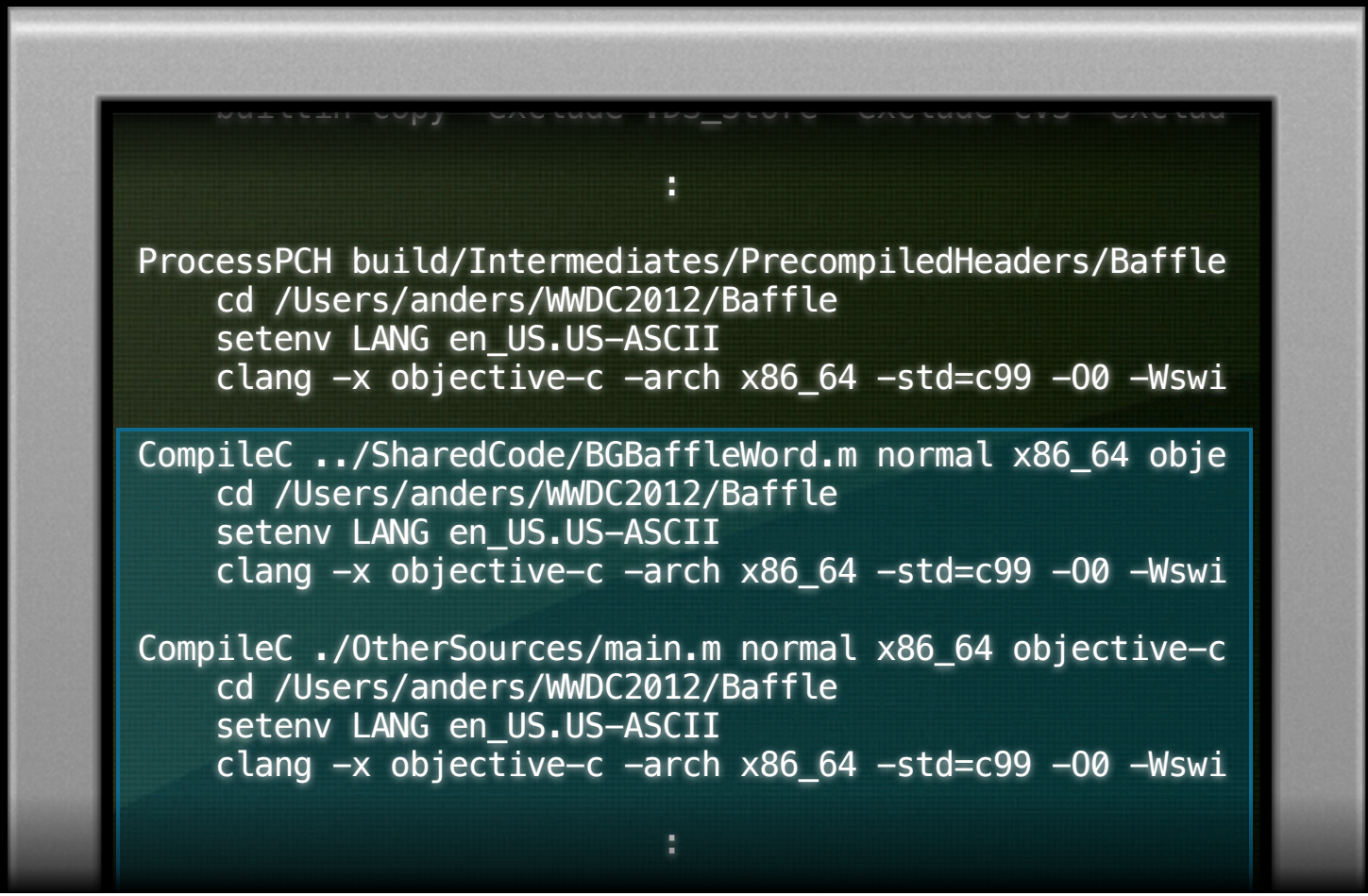
Build Logs in the Console



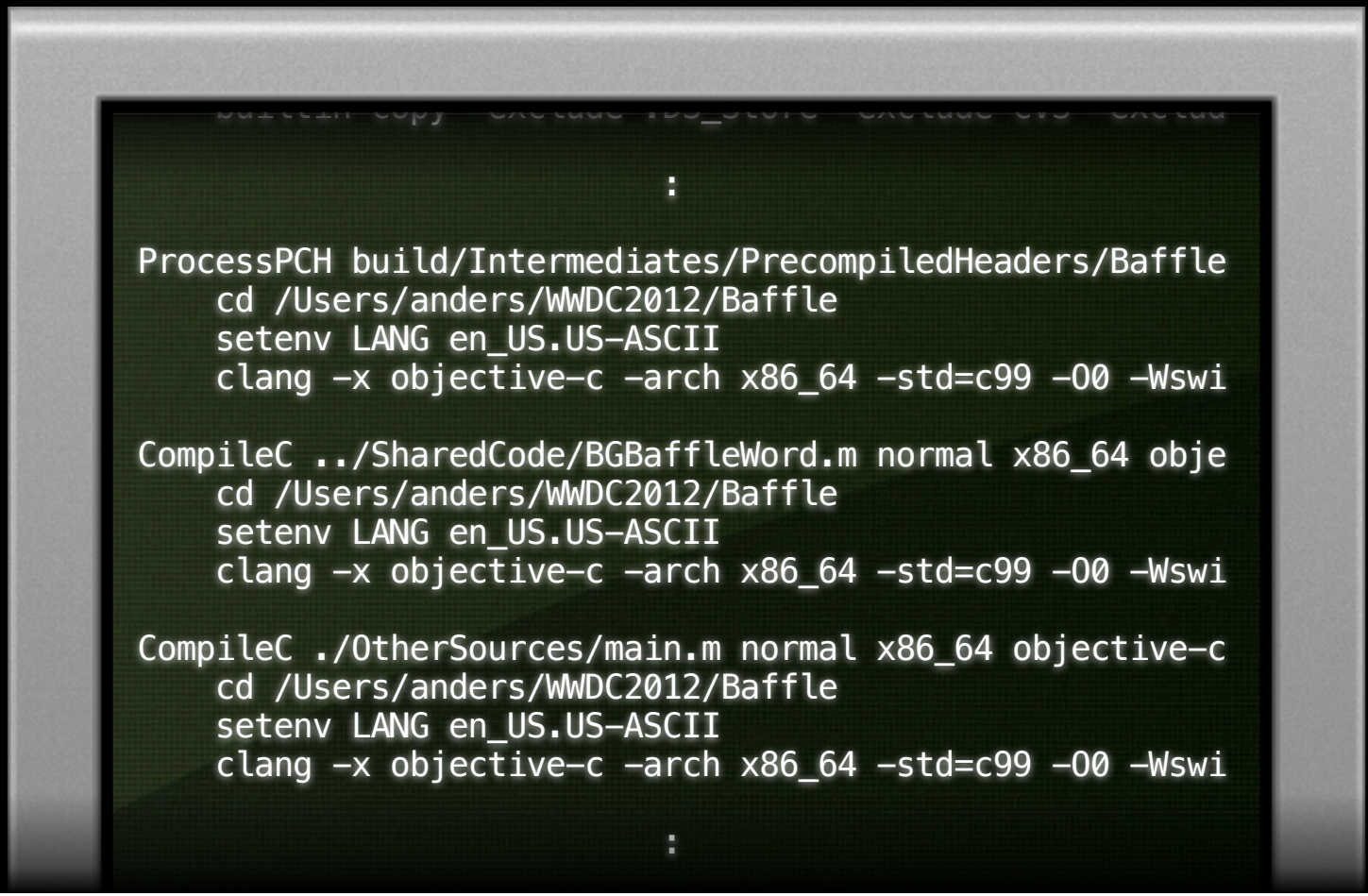
Build Logs in the Console



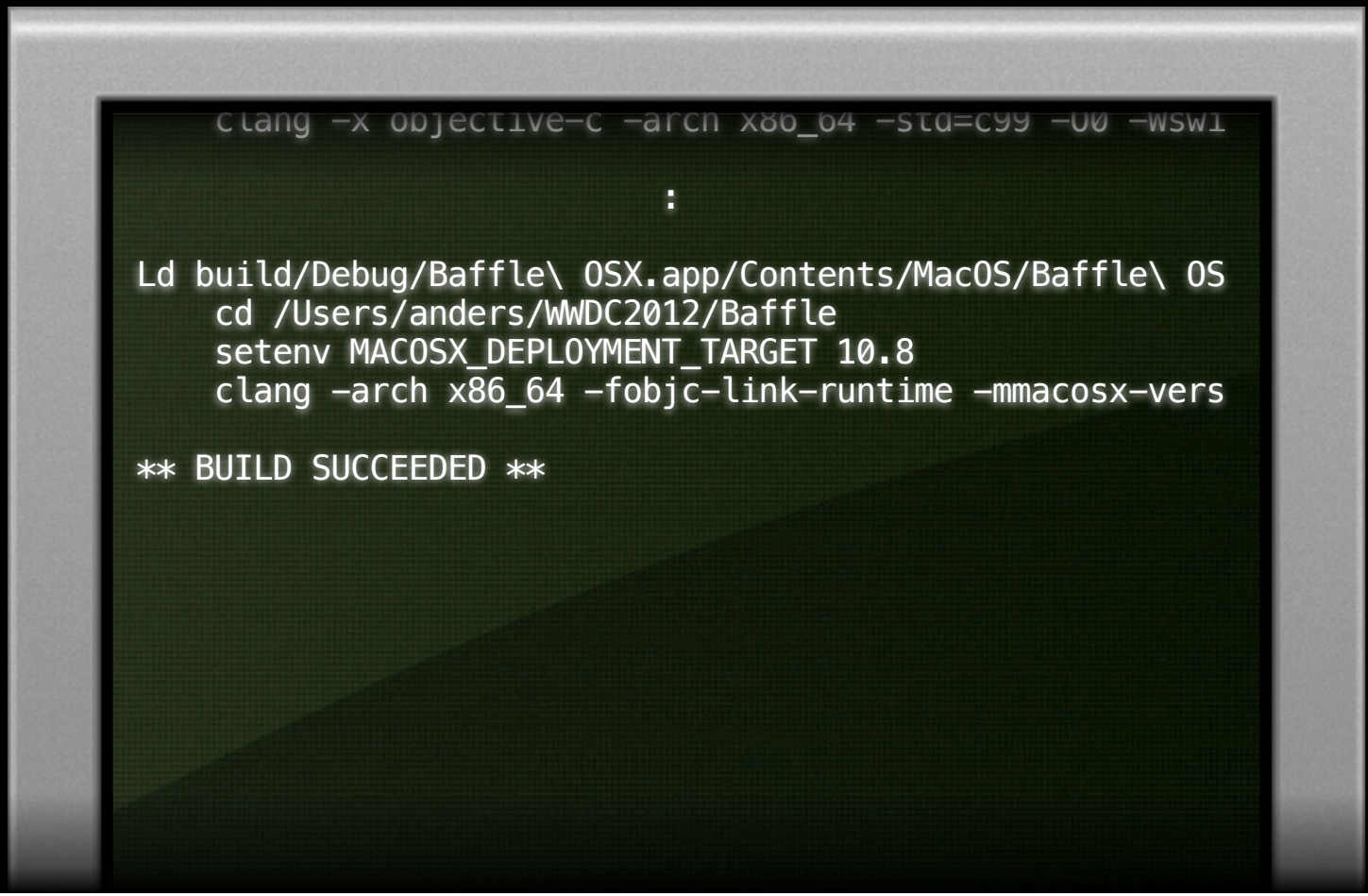
Build Logs in the Console



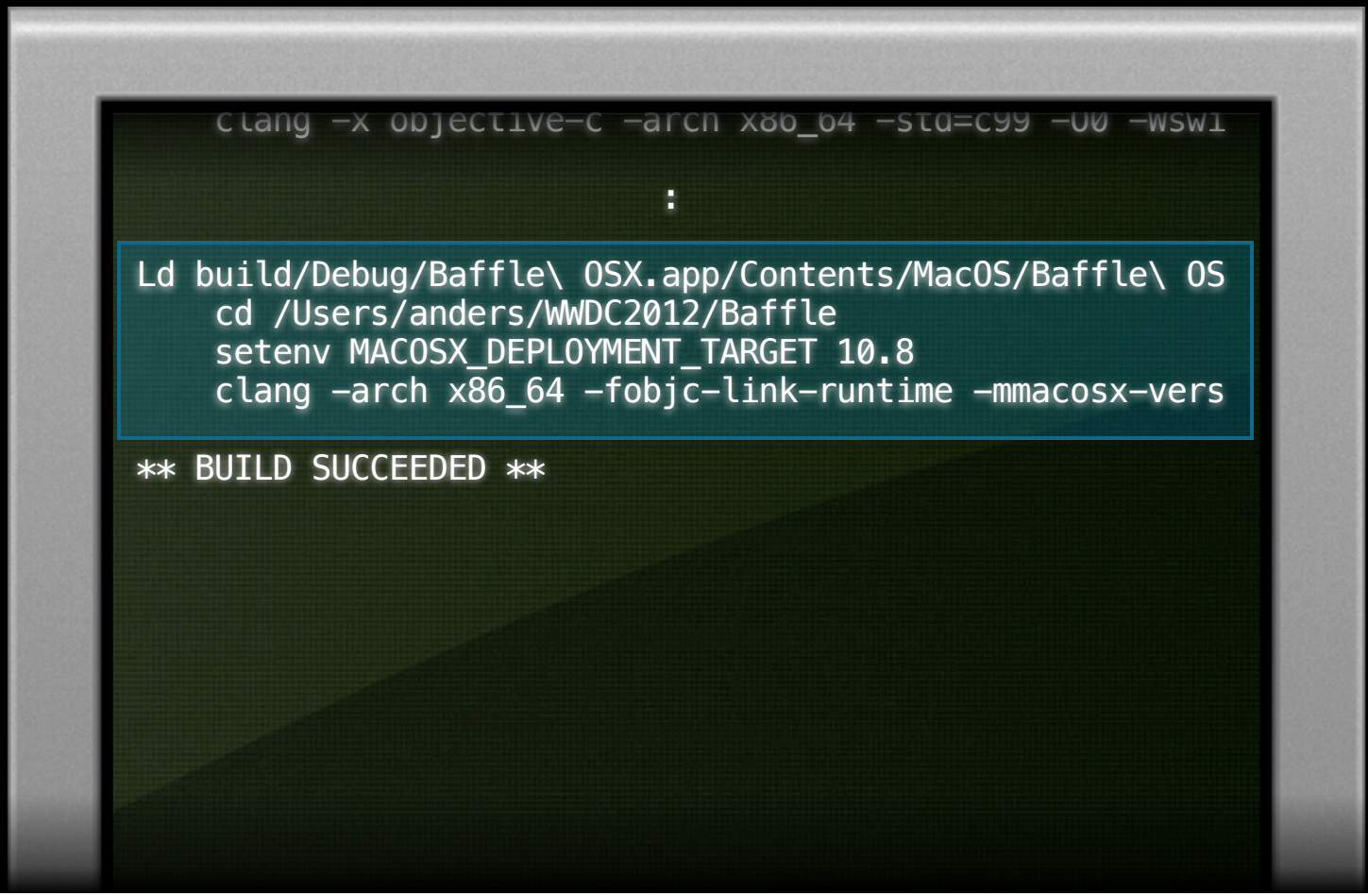
Build Logs in the Console



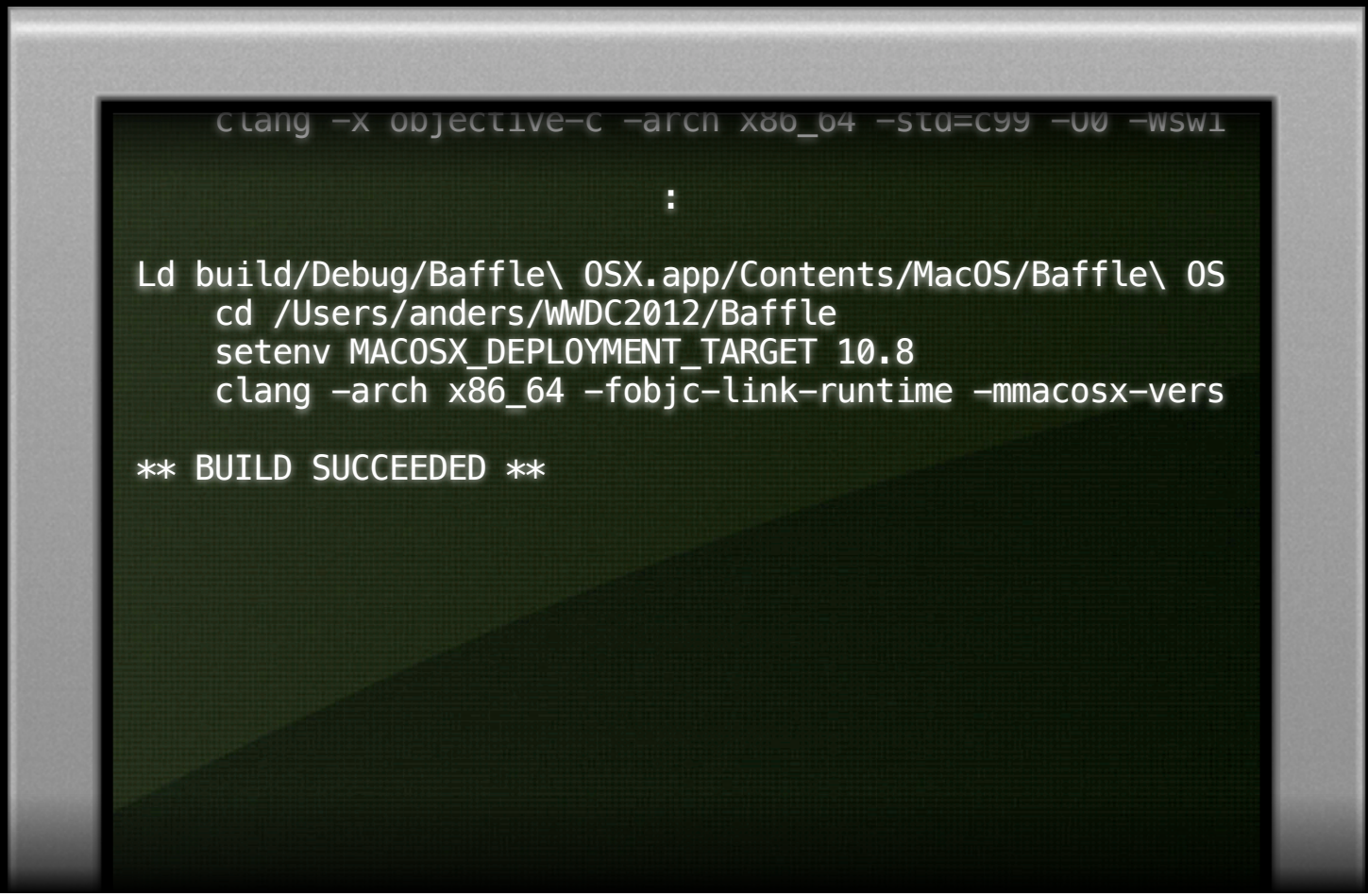
Build Logs in the Console



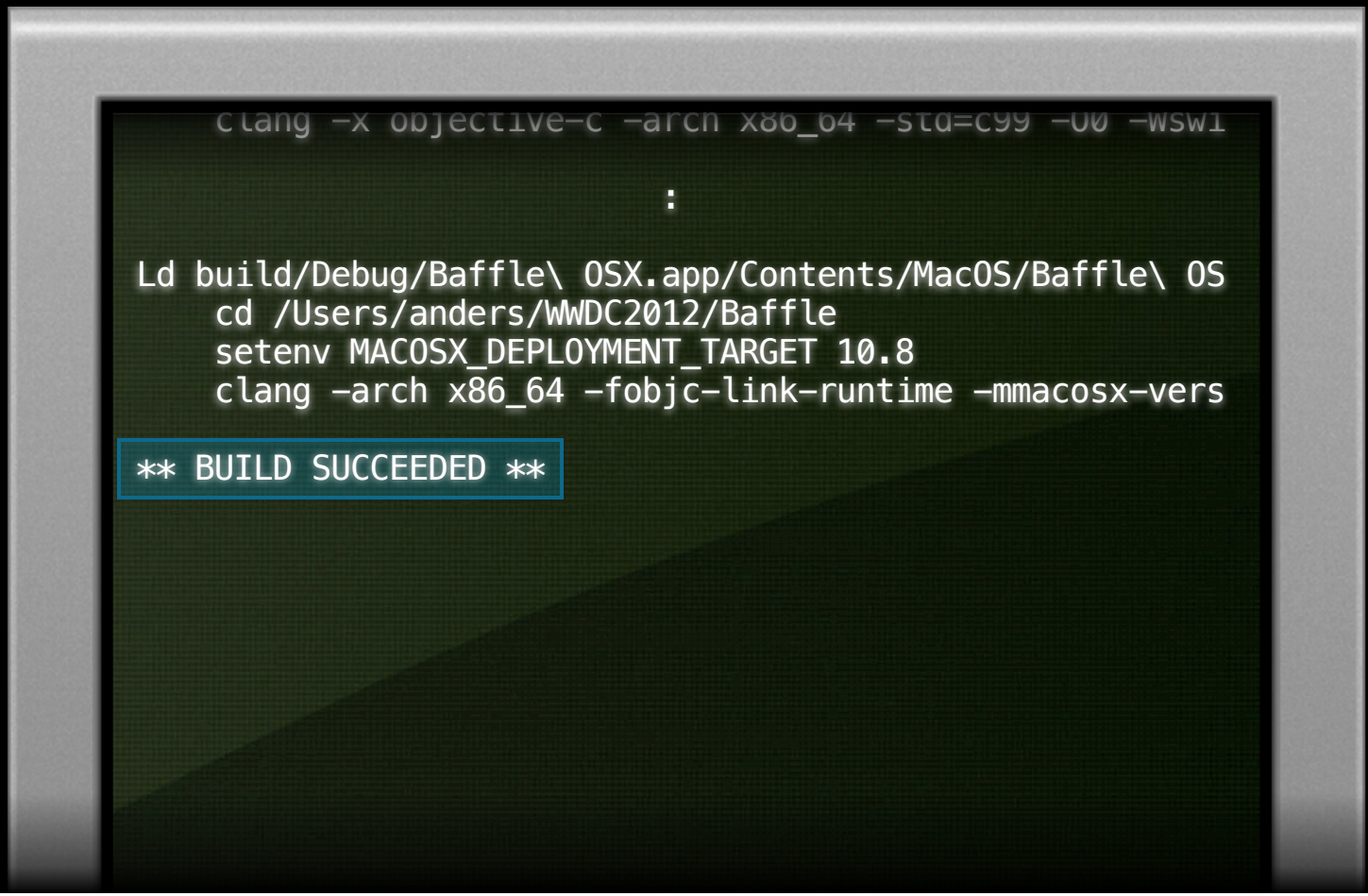
Build Logs in the Console



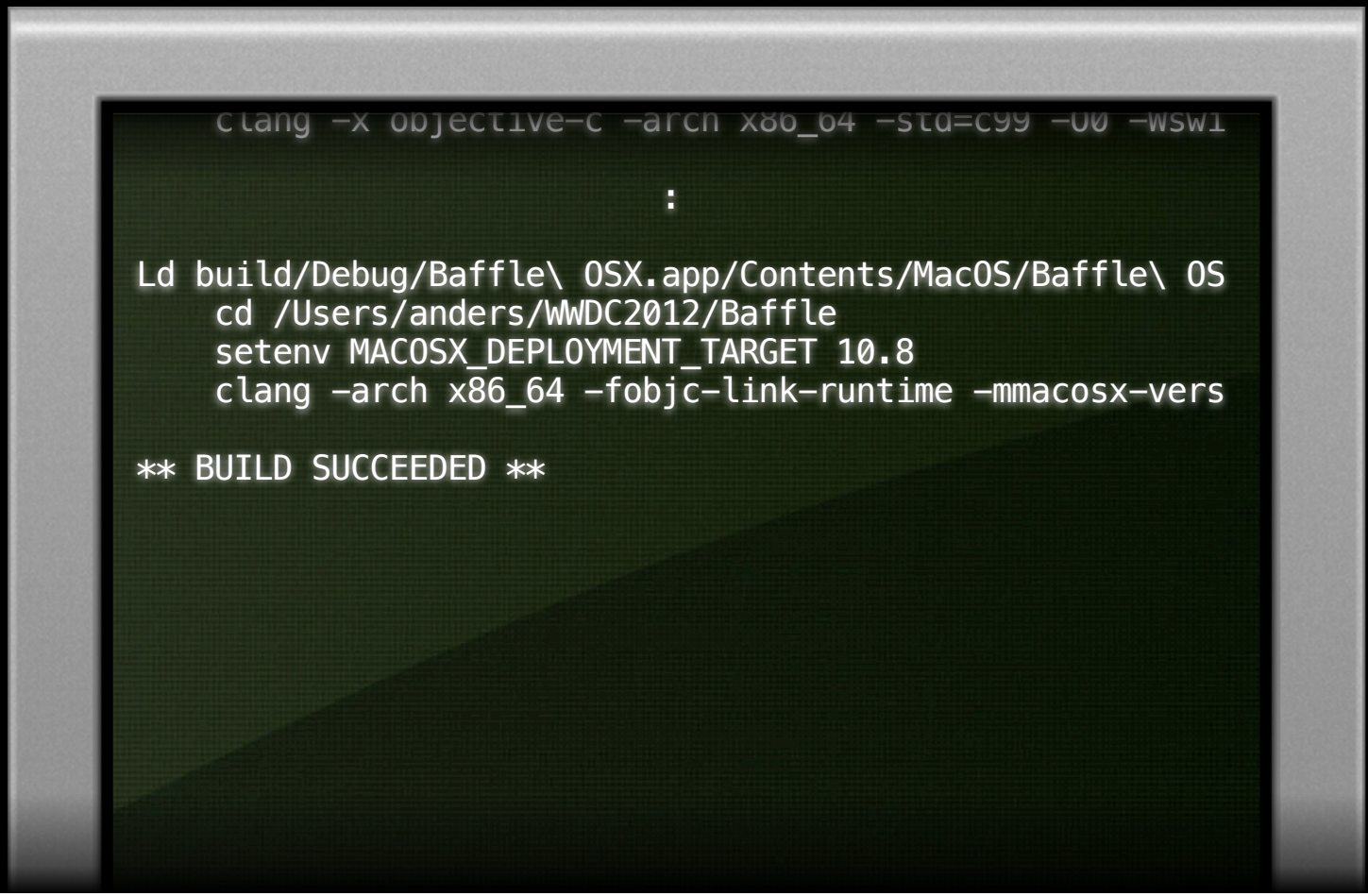
Build Logs in the Console



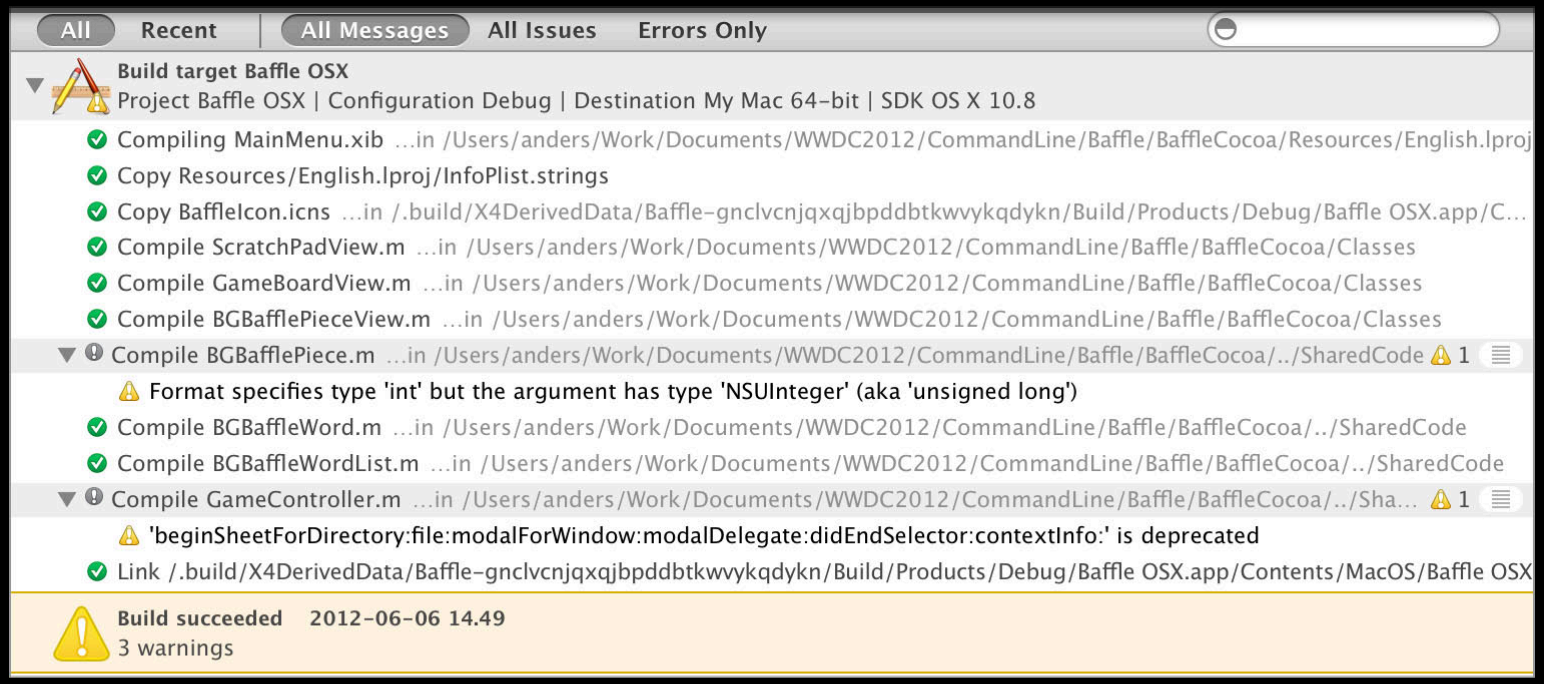
Build Logs in the Console



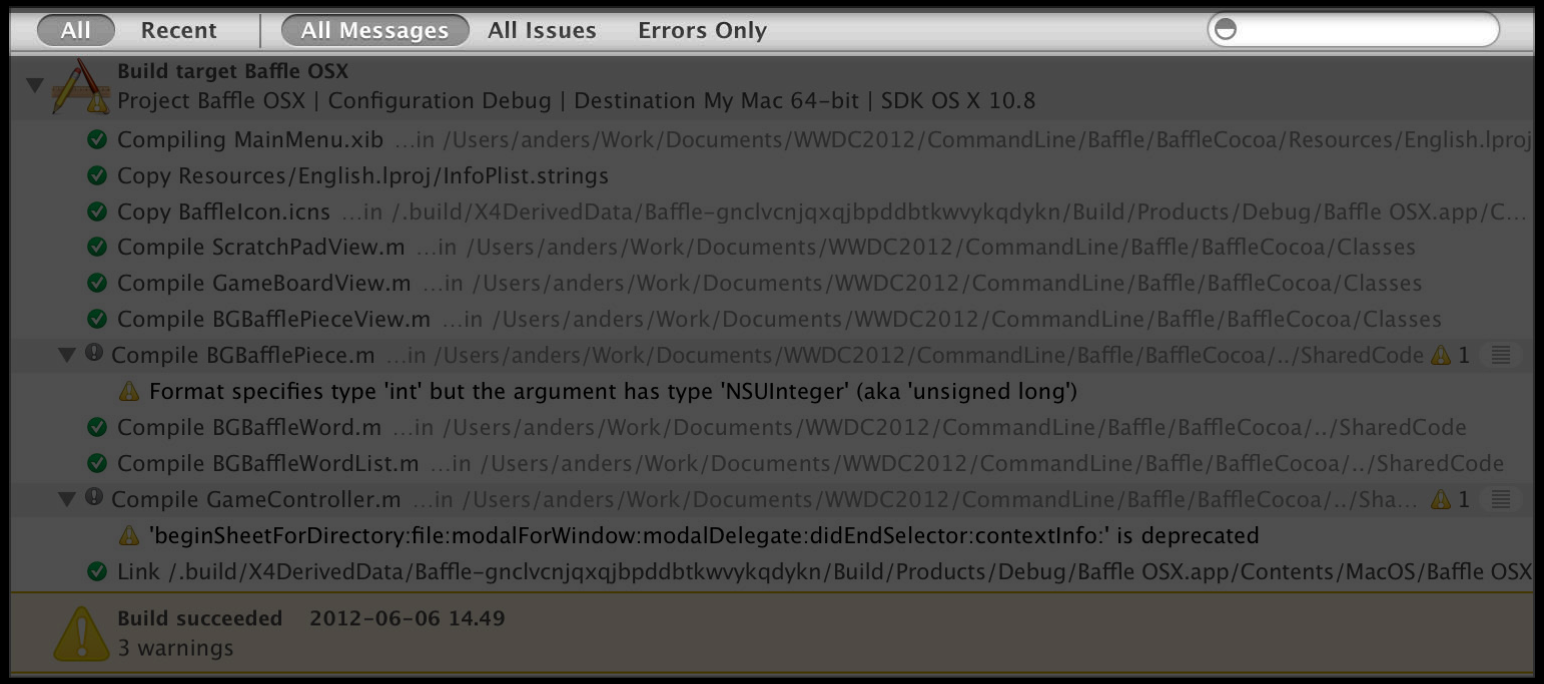
Build Logs in the Console



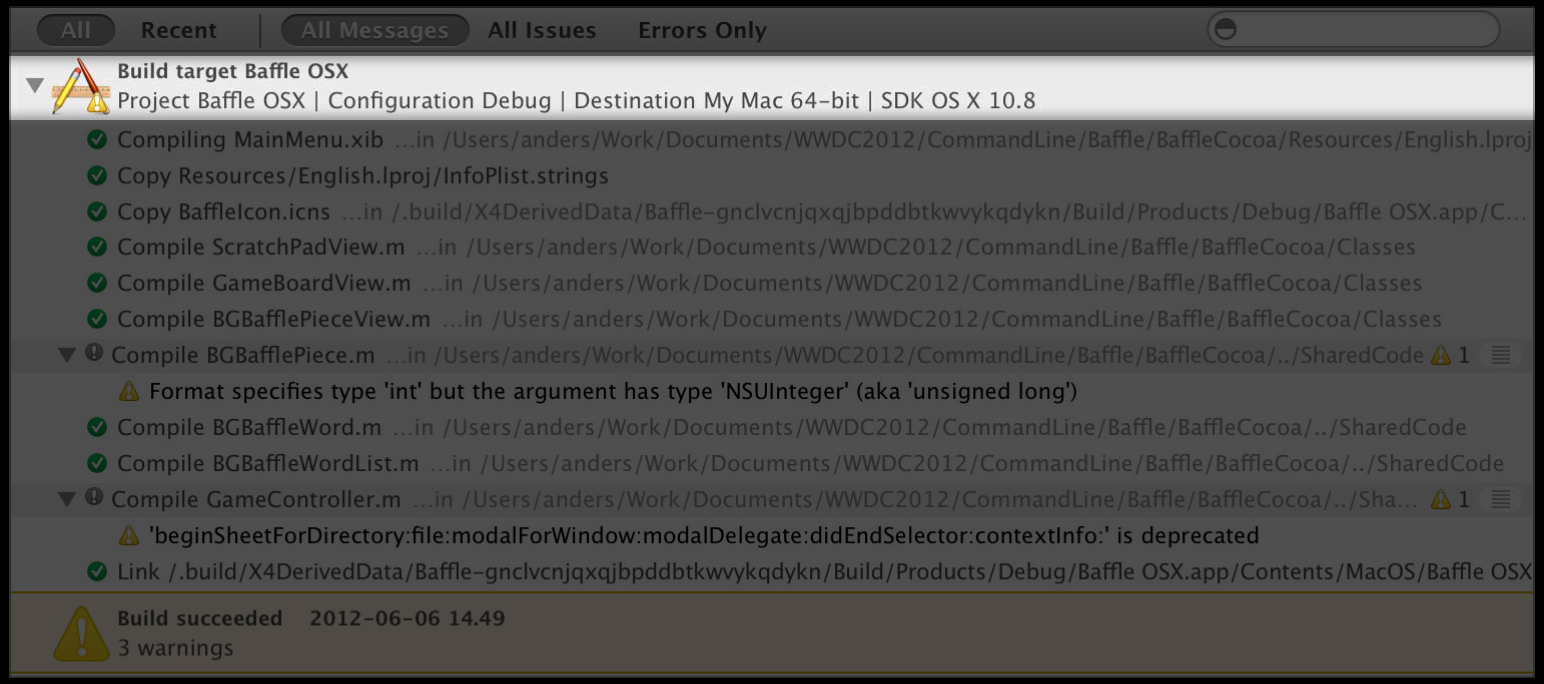
Build Logs in the IDE



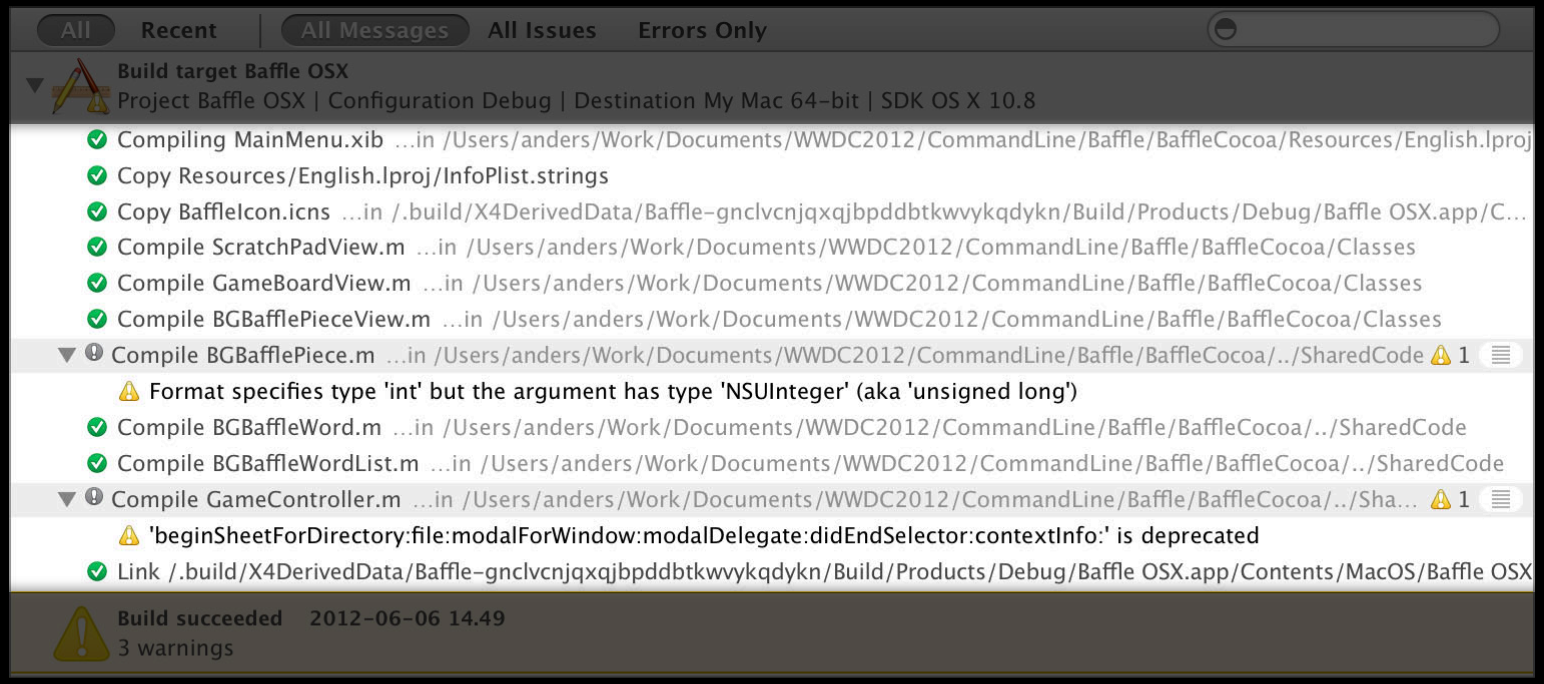
Build Logs in the IDE



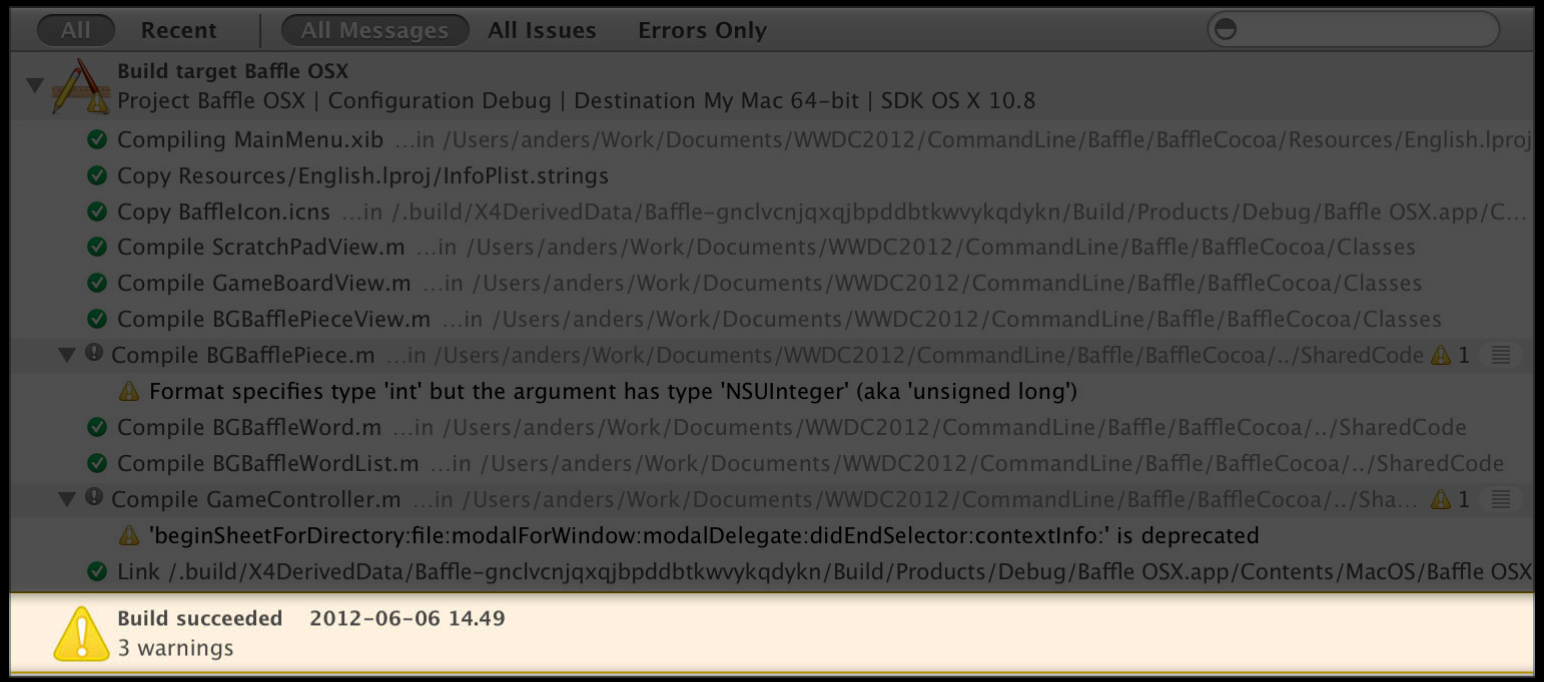
Build Logs in the IDE



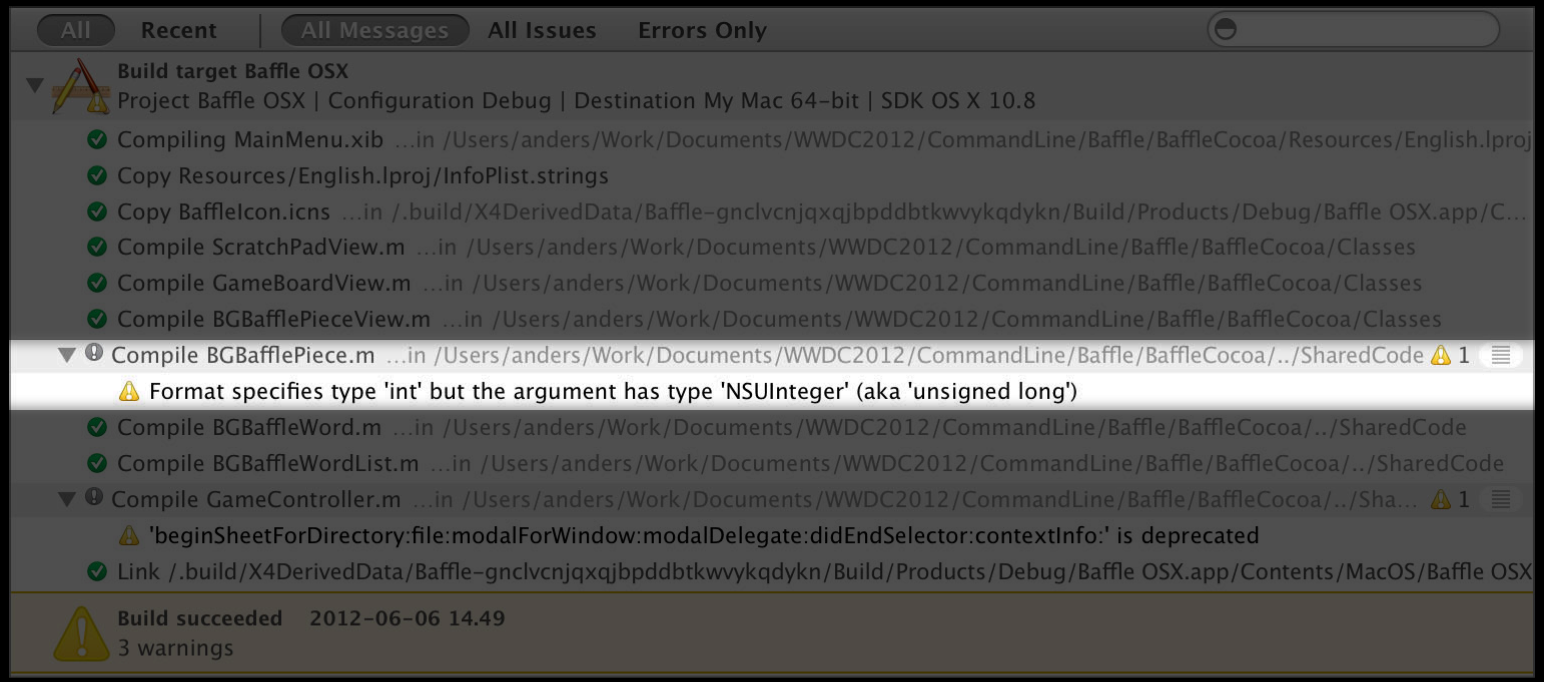
Build Logs in the IDE



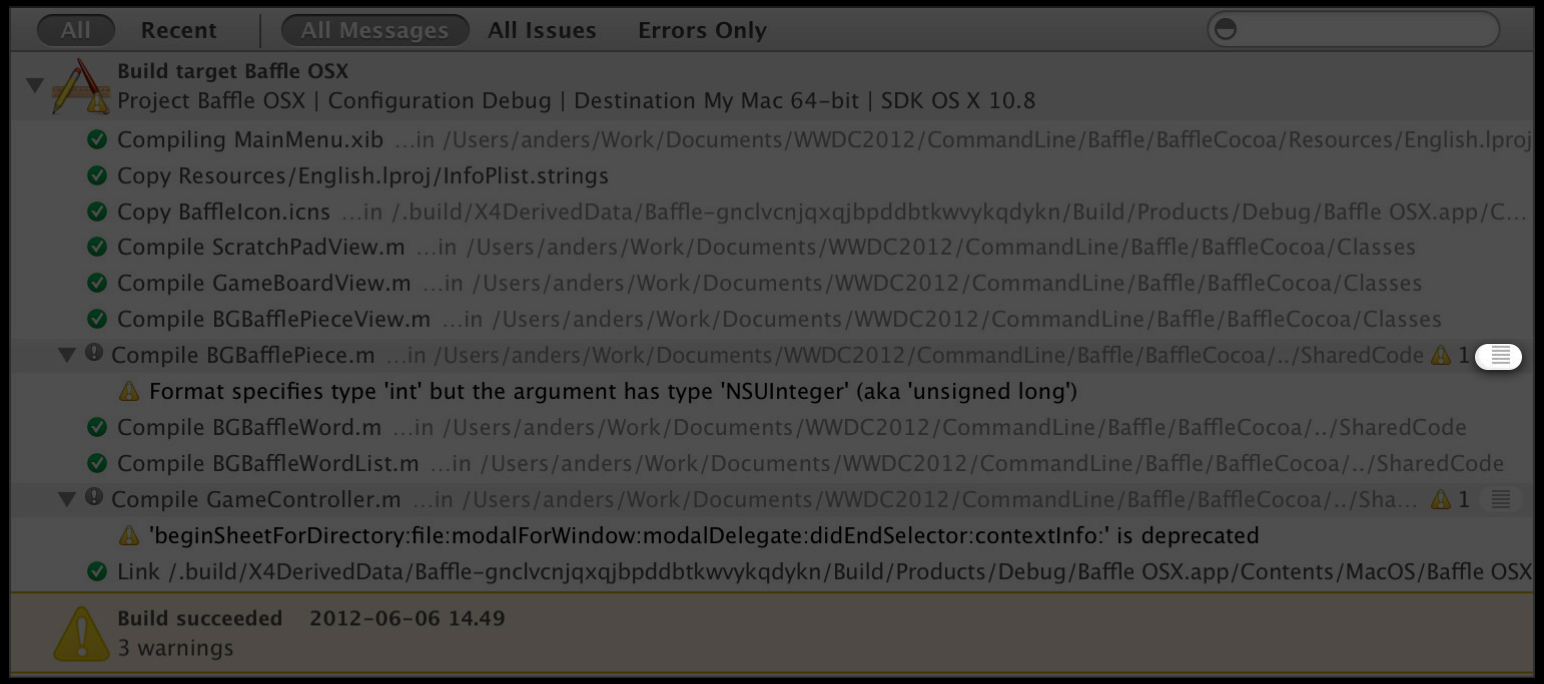
Build Logs in the IDE



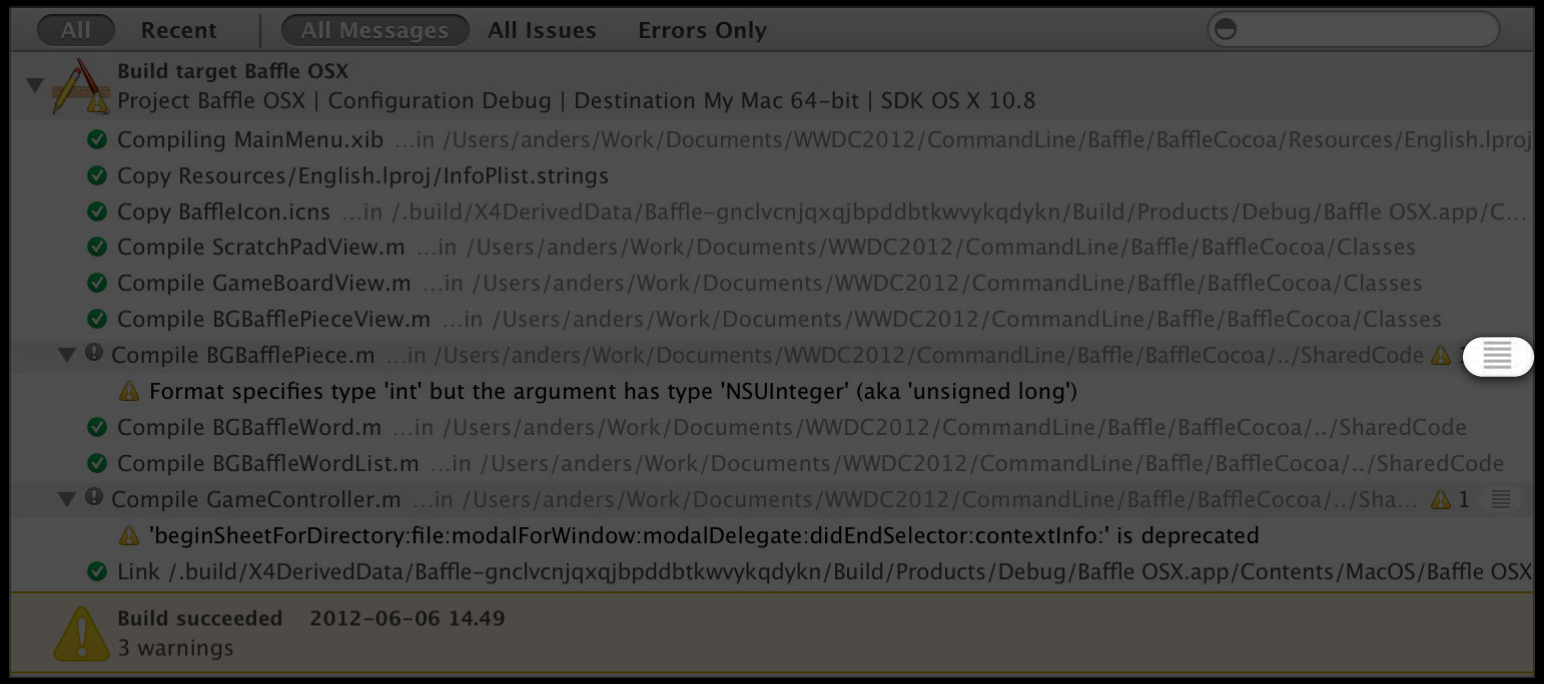
Build Logs in the IDE



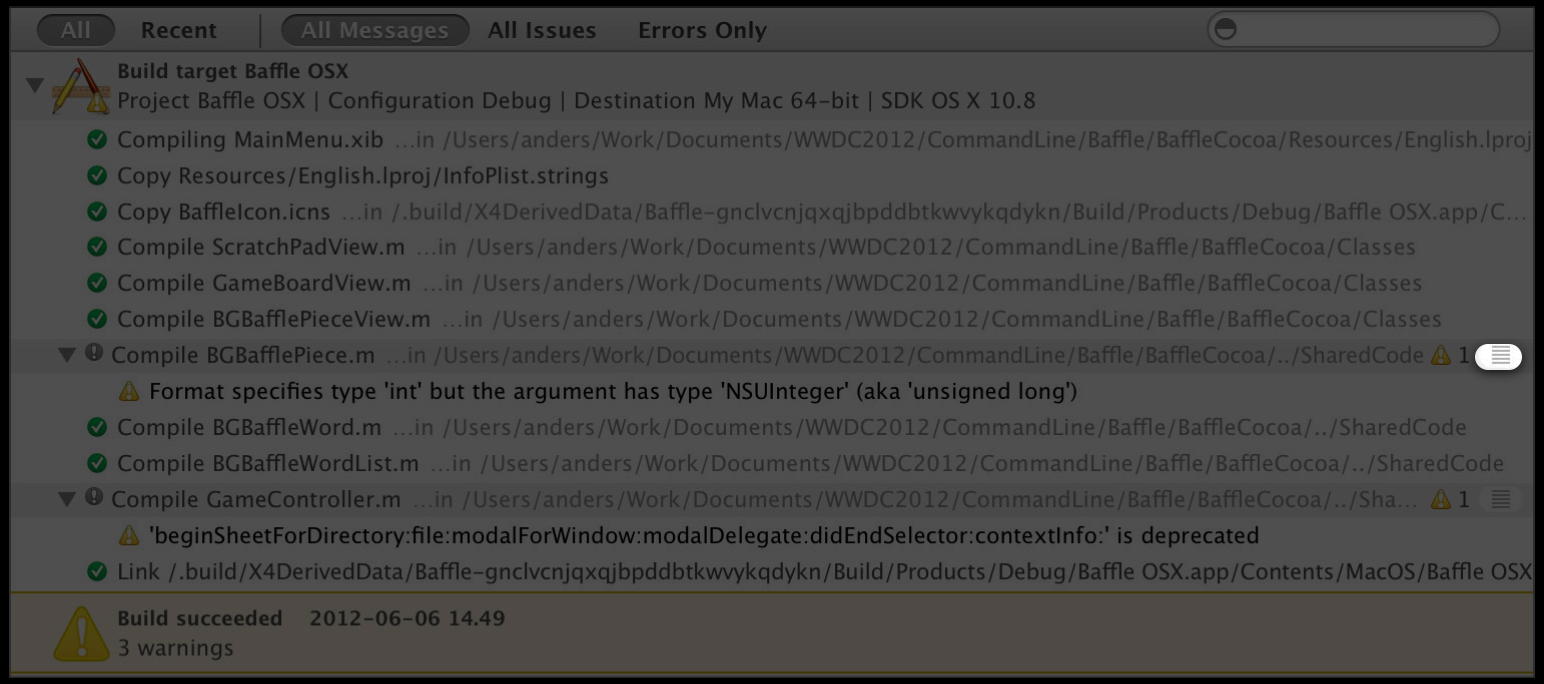
Build Logs in the IDE



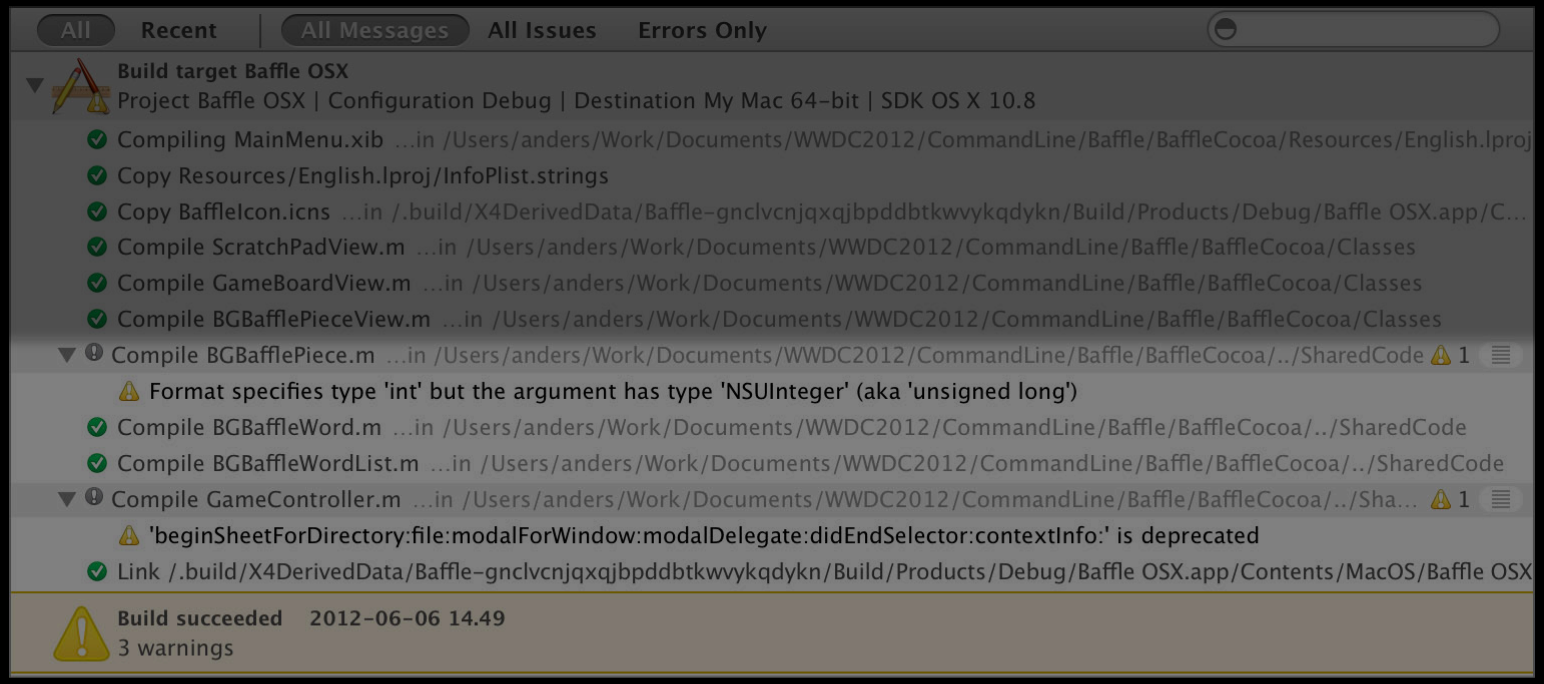
Build Logs in the IDE



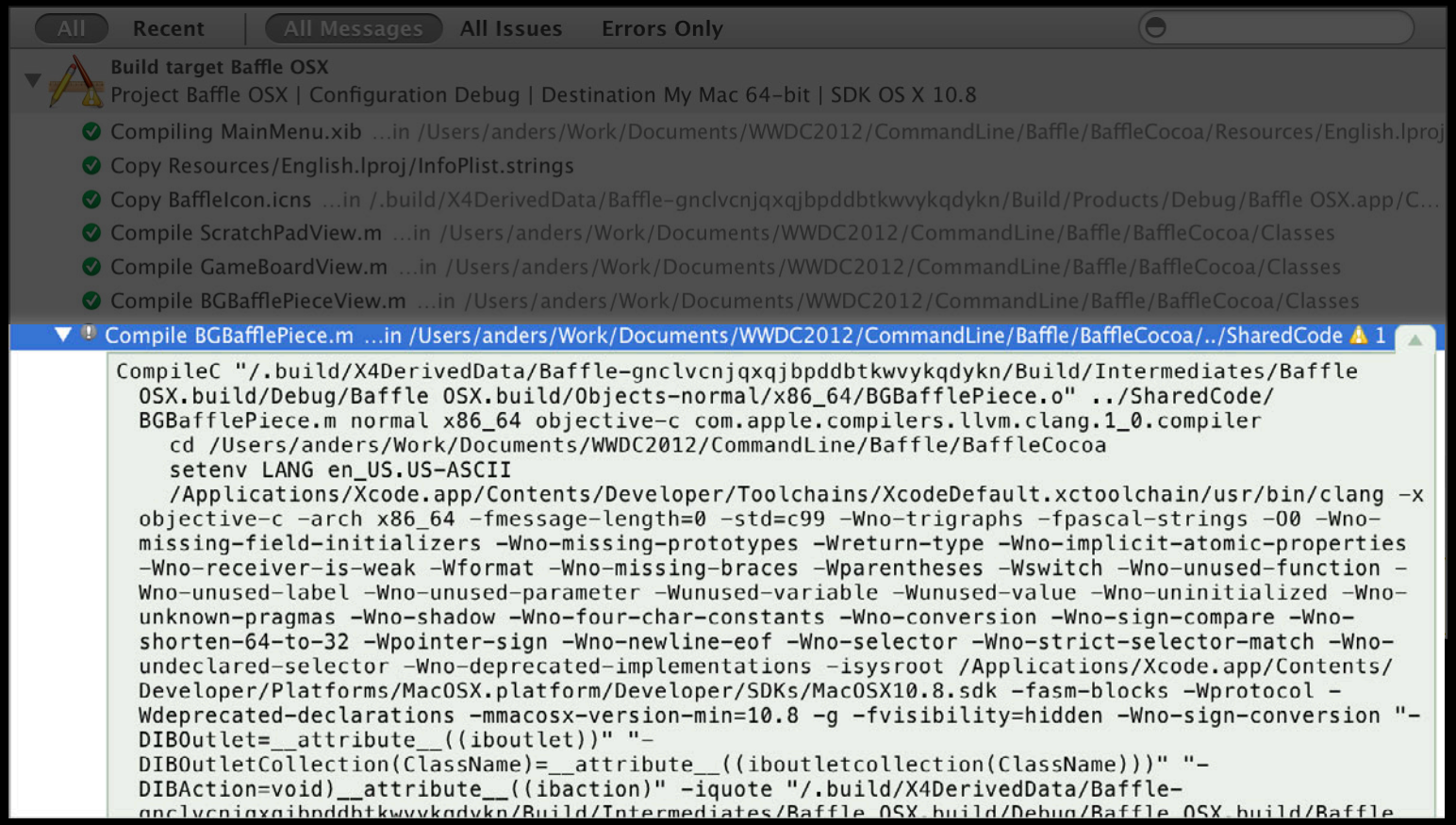
Build Logs in the IDE



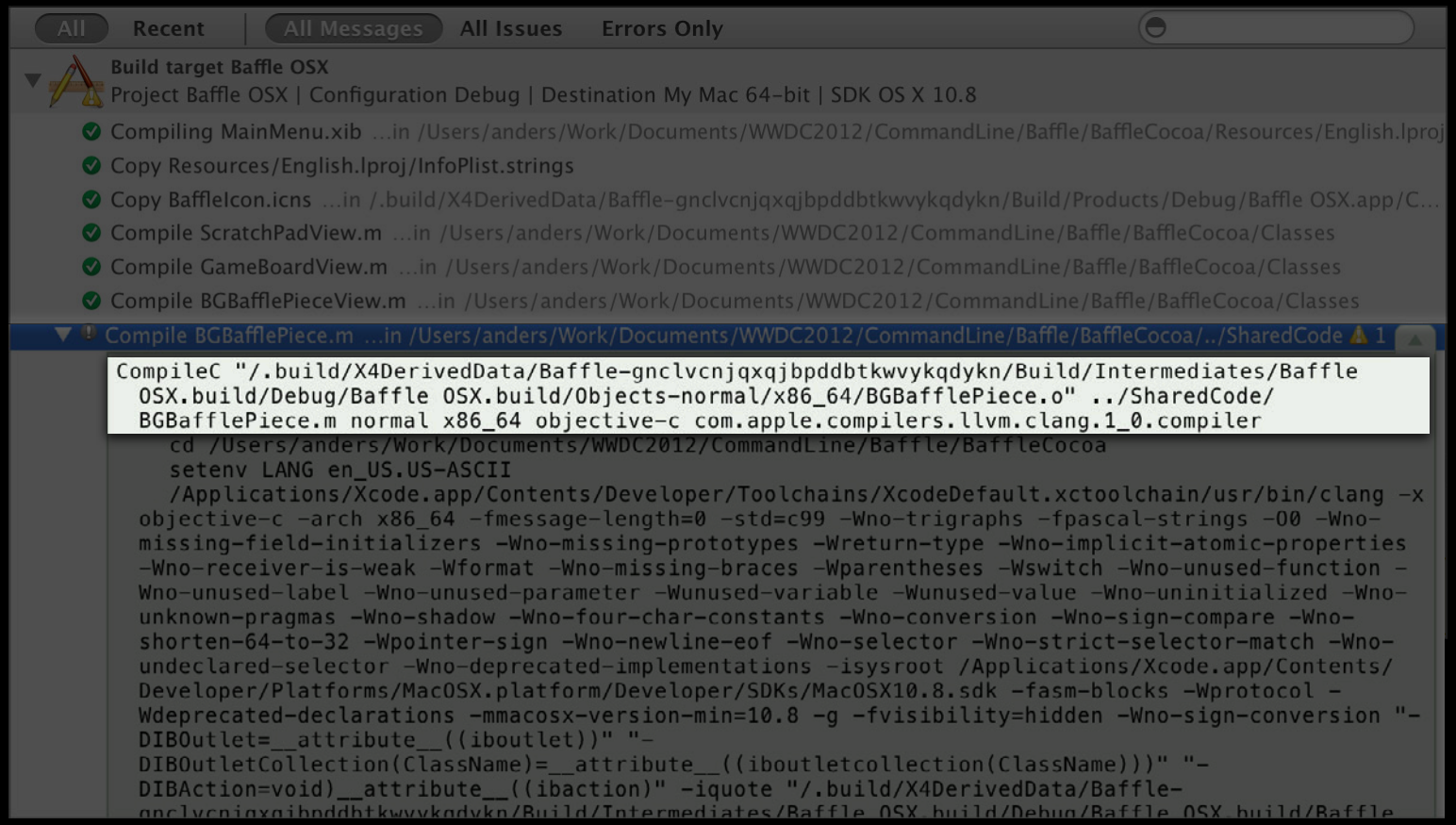
Build Logs in the IDE



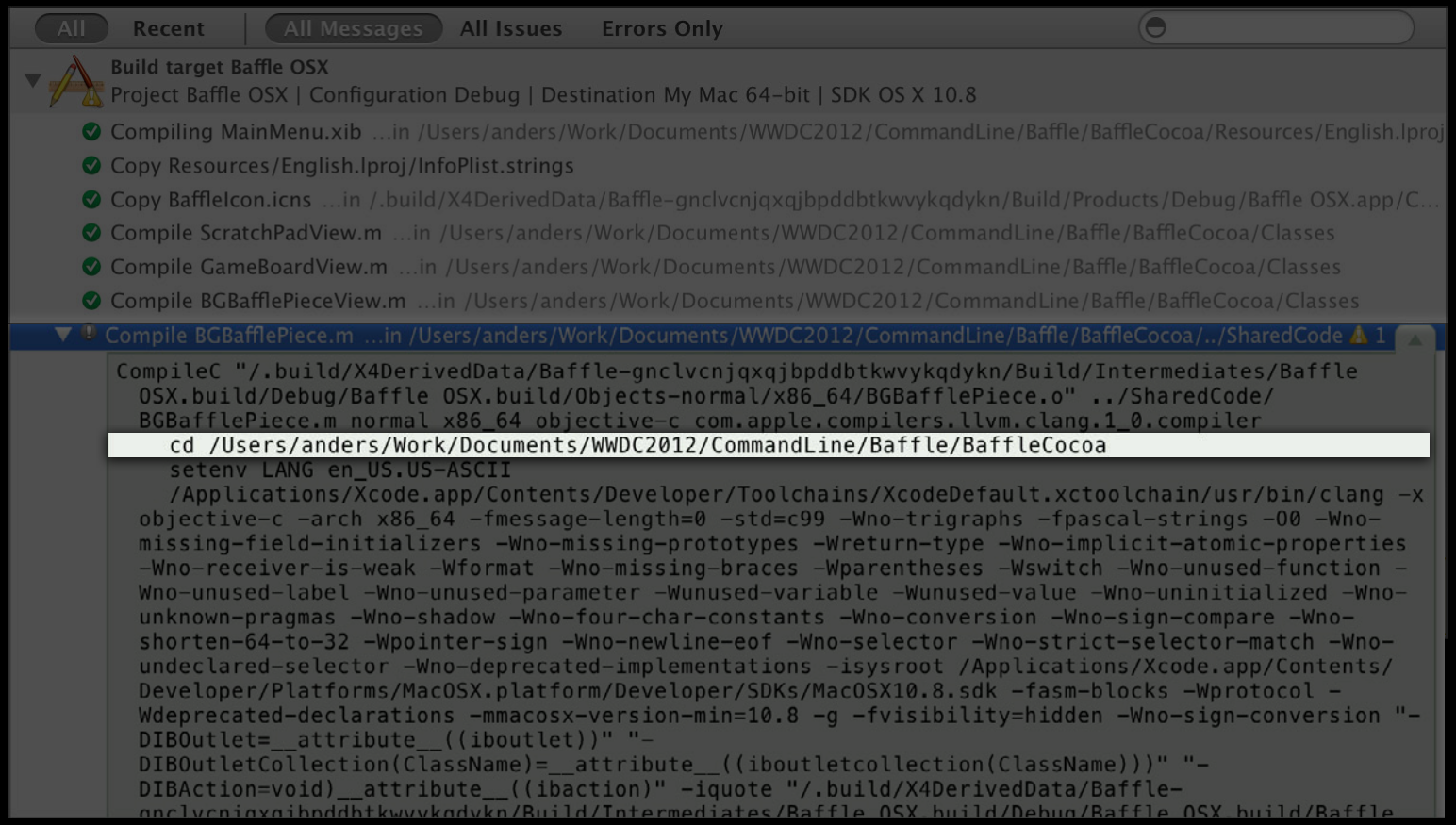
Build Logs in the IDE



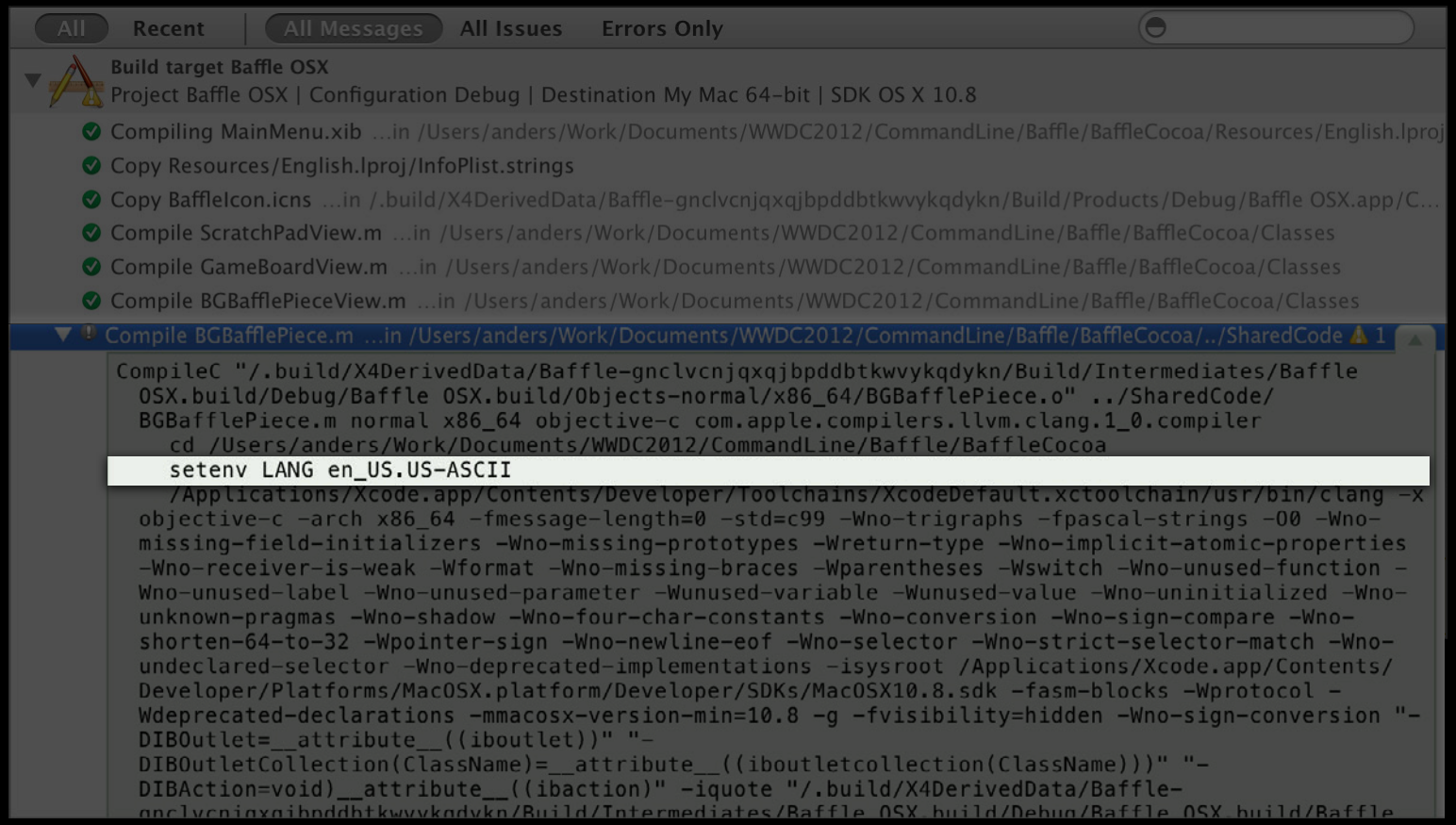
Build Logs in the IDE



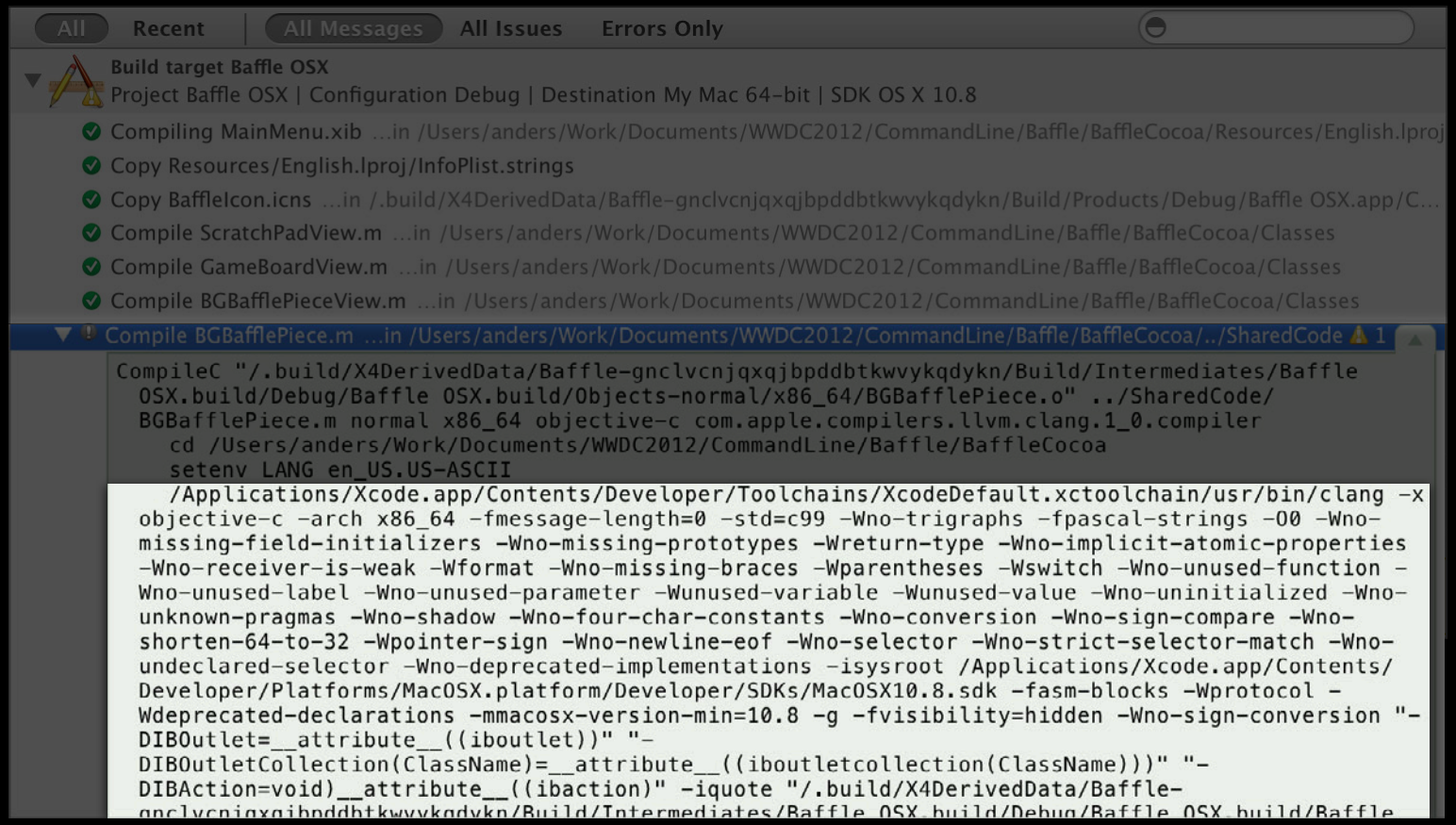
Build Logs in the IDE



Build Logs in the IDE



Build Logs in the IDE



Where Do the Outputs Go?

Kind	Contents	Controlling Build Setting
Intermediates	Compiled Code (“Objects”)	OBJROOT
	Generated Sources	
	Precompiled Headers	SHARED_PRECOMPS_DIR
Debug Products	Products, in debug locations	SYMROOT
	Debug Symbols (.dSYMs)	
Release Products	Products, in release locations	DSTROOT

Where Do the Outputs Go?

Kind	Contents	Controlling Build Setting
Intermediates	Compiled Code (“Objects”)	OBJROOT
	Generated Sources	
	Precompiled Headers	SHARED_PRECOMPS_DIR
Debug Products	Products, in debug locations	SYMROOT
	Debug Symbols (.dSYMs)	
Release Products	Products, in release locations	DSTROOT

Where Do the Outputs Go?

Kind	Contents	Controlling Build Setting
Intermediates	Compiled Code (“Objects”)	OBJROOT
	Generated Sources	
	Precompiled Headers	SHARED_PRECOMPS_DIR
Debug Products	Products, in debug locations	SYMROOT
	Debug Symbols (.dSYMs)	
Release Products	Products, in release locations	DSTROOT

Where Do the Outputs Go?

Kind	Contents	Controlling Build Setting
Intermediates	Compiled Code (“Objects”)	OBJROOT
	Generated Sources	
	Precompiled Headers	SHARED_PRECOMPS_DIR
Debug Products	Products, in debug locations	SYMROOT
	Debug Symbols (.dSYMs)	
Release Products	Products, in release locations	DSTROOT

Customizing Your Project

Controlling the Build

- Schemes

Controlling the Build

- Schemes
- Overriding build settings

Controlling the Build

- Schemes
- Overriding build settings
  - Configuration files

Controlling the Build

- Schemes
- Overriding build settings
  - Configuration files
  - Architectures

Controlling the Build

- Schemes
- Overriding build settings
  - Configuration files
  - Architectures
  - SDKs

Controlling the Build

- Schemes
- Overriding build settings
  - Configuration files
  - Architectures
  - SDKs
  - Other build settings

Schemes

- A scheme defines:

Schemes

- A scheme defines:
  - A collection of targets to build

Schemes

- A scheme defines:
  - A collection of targets to build
  - A configuration to use when building

Schemes

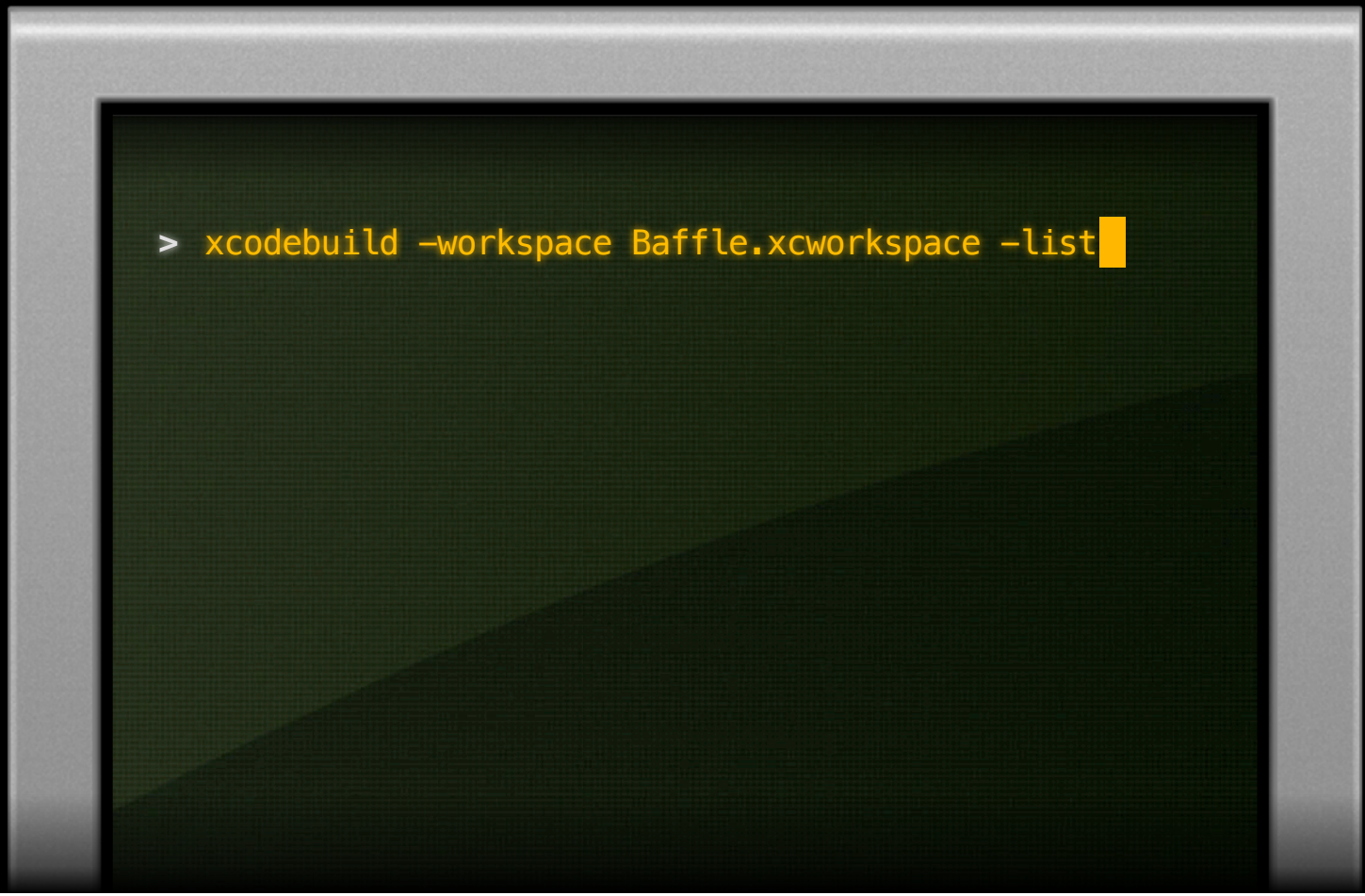
- A scheme defines:
  - A collection of targets to build
  - A configuration to use when building
  - An optional collection of tests to execute

Schemes

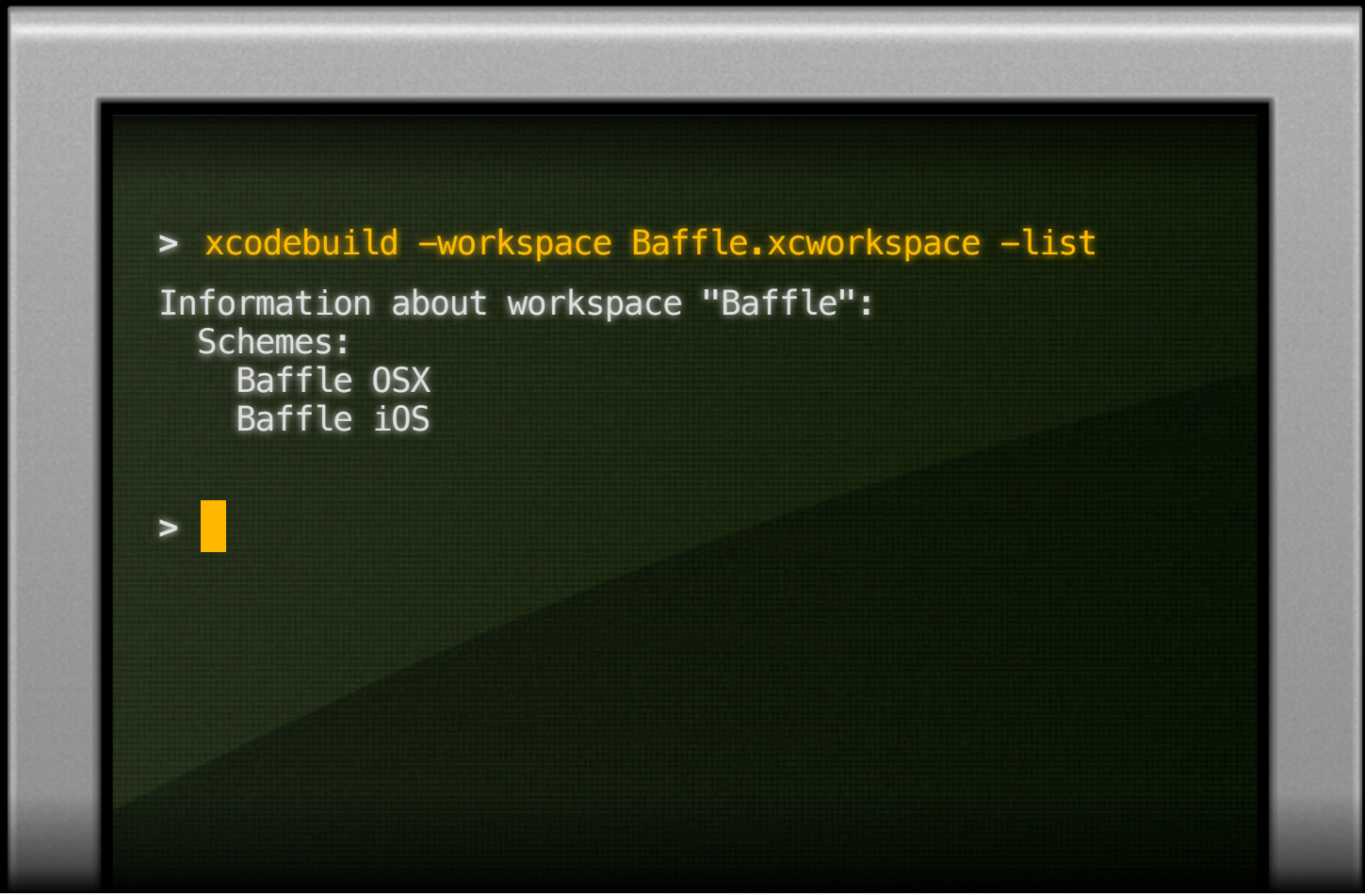
- A scheme defines:
  - A collection of targets to build
  - A configuration to use when building
  - An optional collection of tests to execute
  - User-defined pre- and post-processing actions

Listing the Schemes

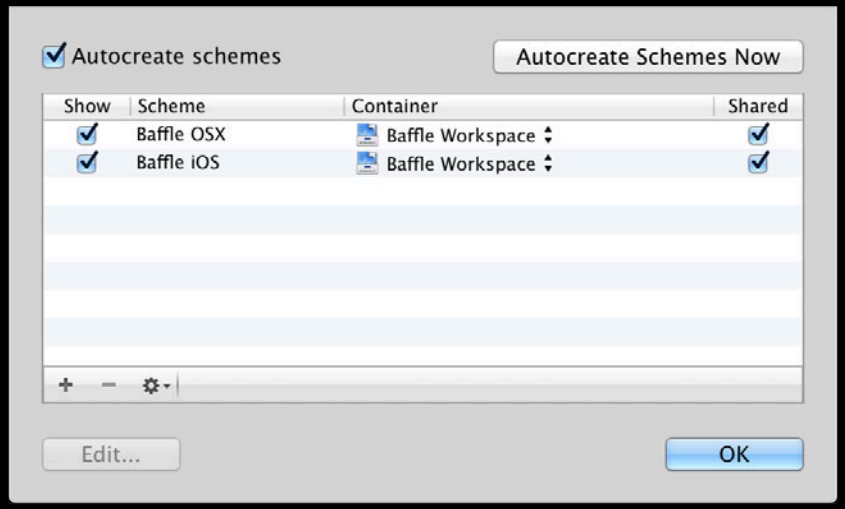
Listing the Schemes



Listing the Schemes



Listing the Schemes



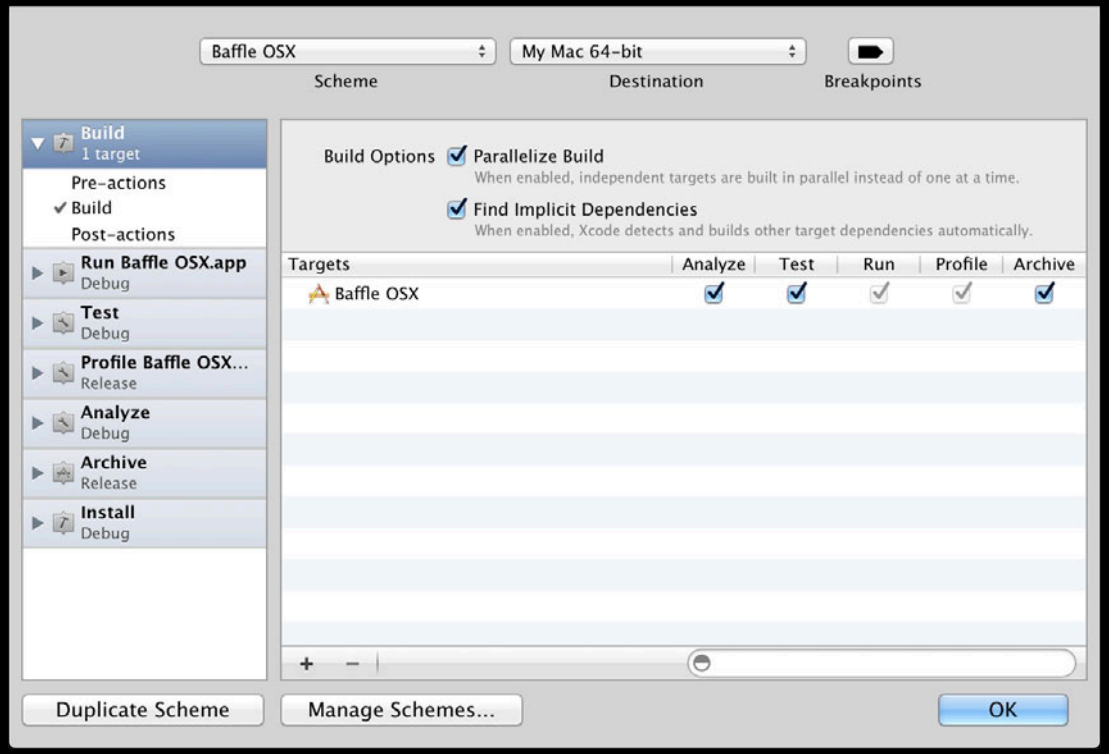
Using Schemes

- Configure in Xcode
  - Build configuration
  - Scheme actions
  - Other options

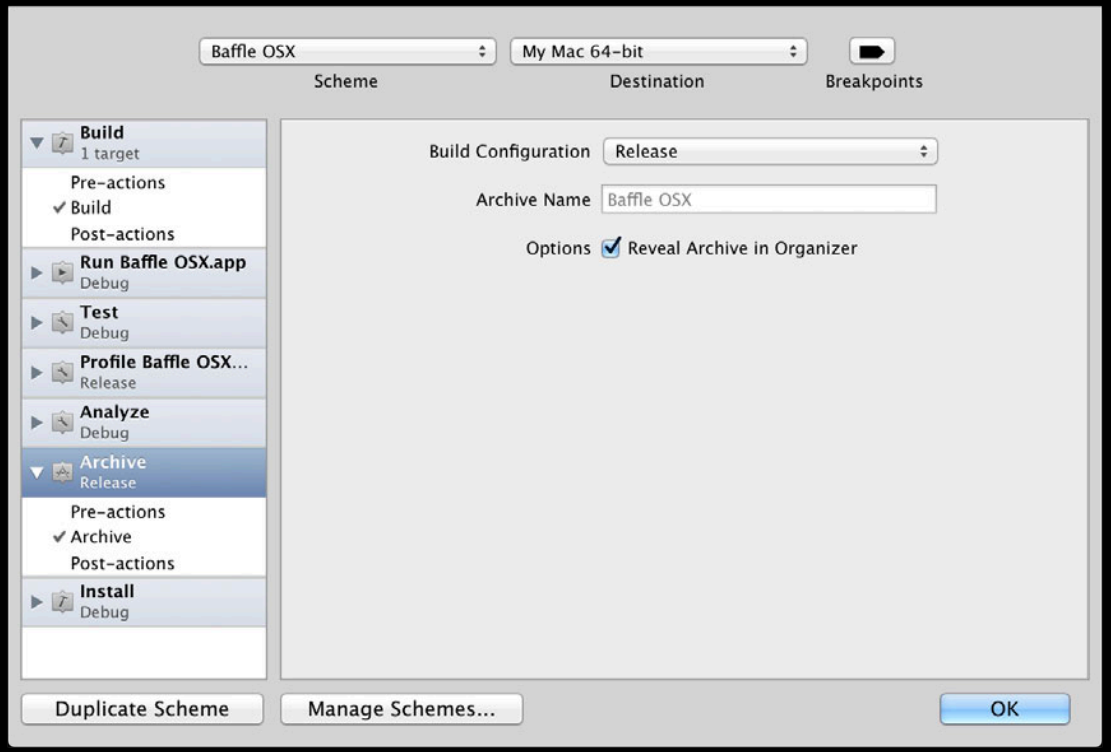
Using Schemes

- Configure in Xcode
  - Build configuration
  - Scheme actions
  - Other options
- Use in `xcodebuild`
  - Use `-scheme` flag to select scheme to use

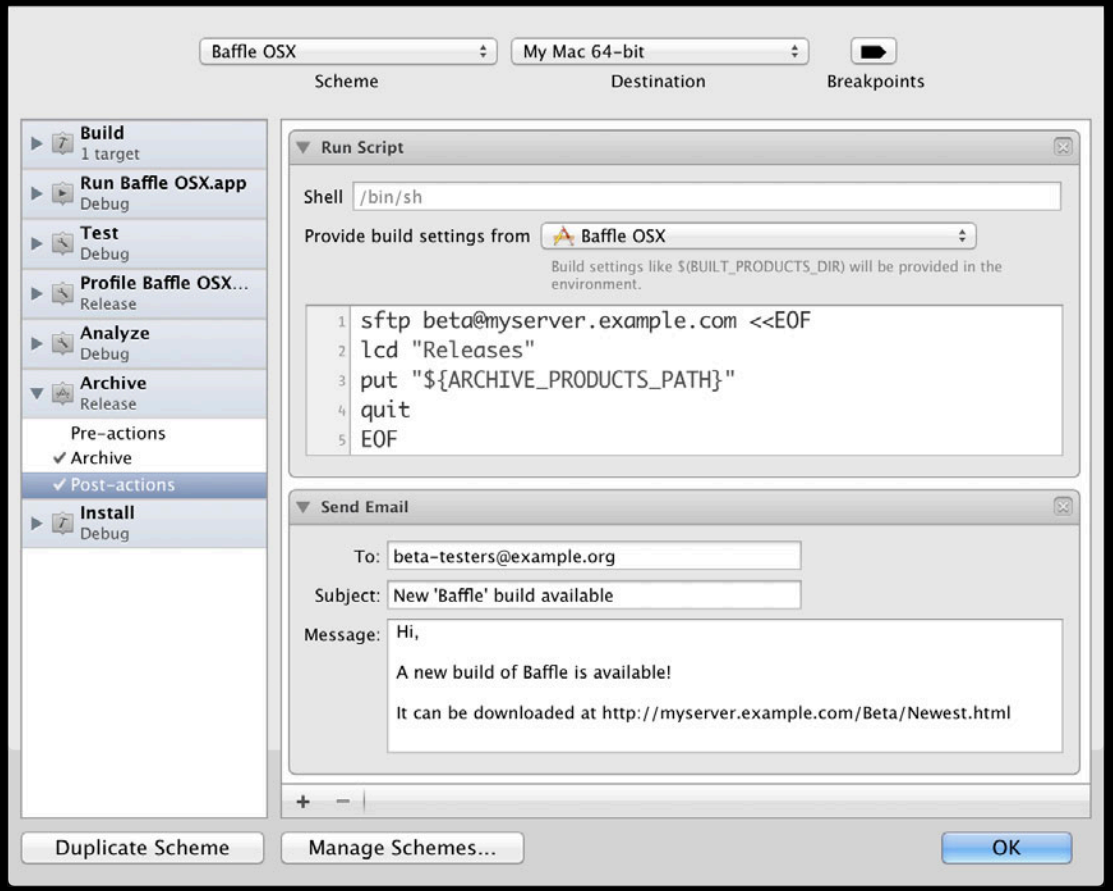
Customizing Schemes



Customizing Schemes

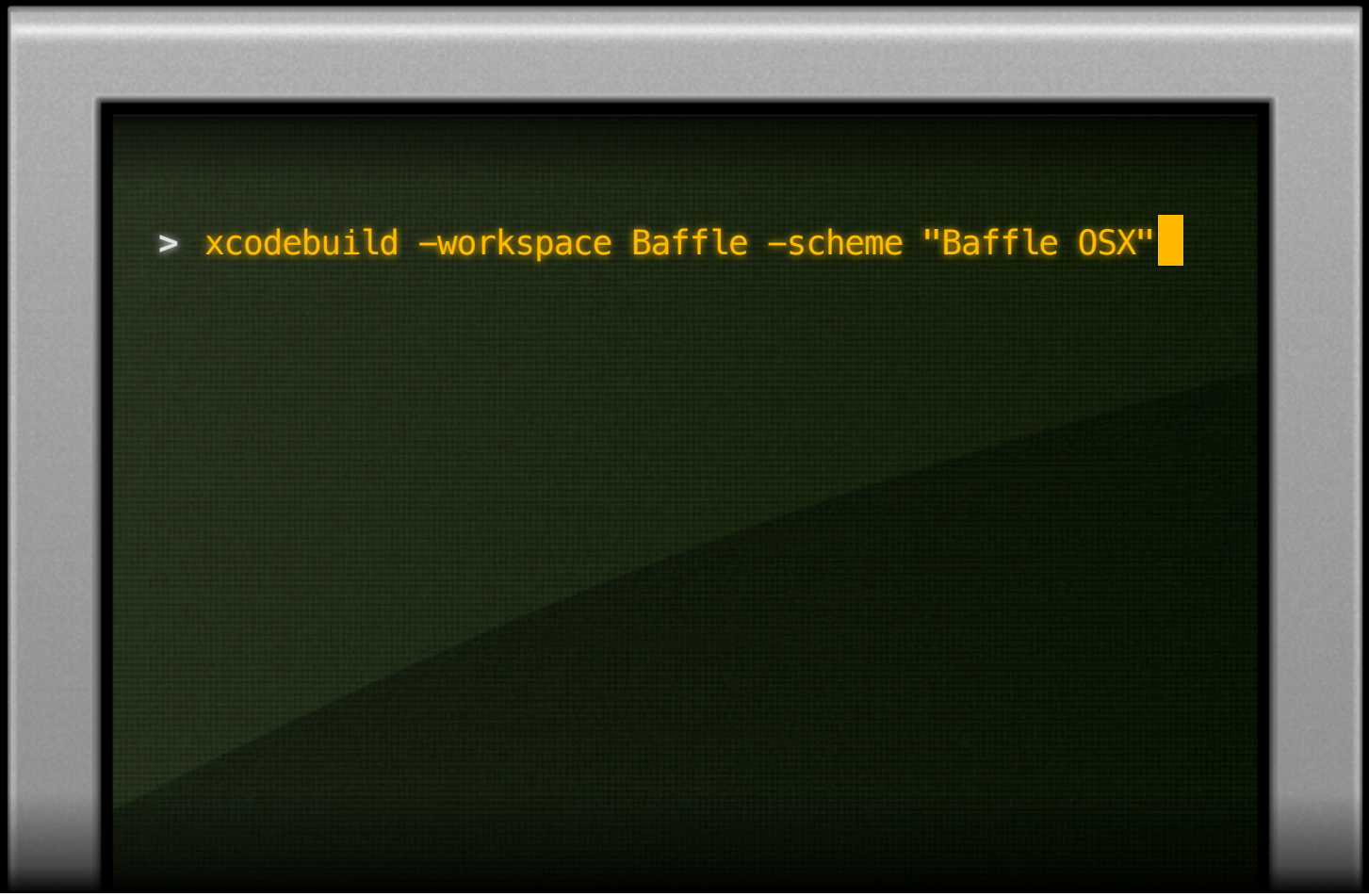


Customizing Schemes



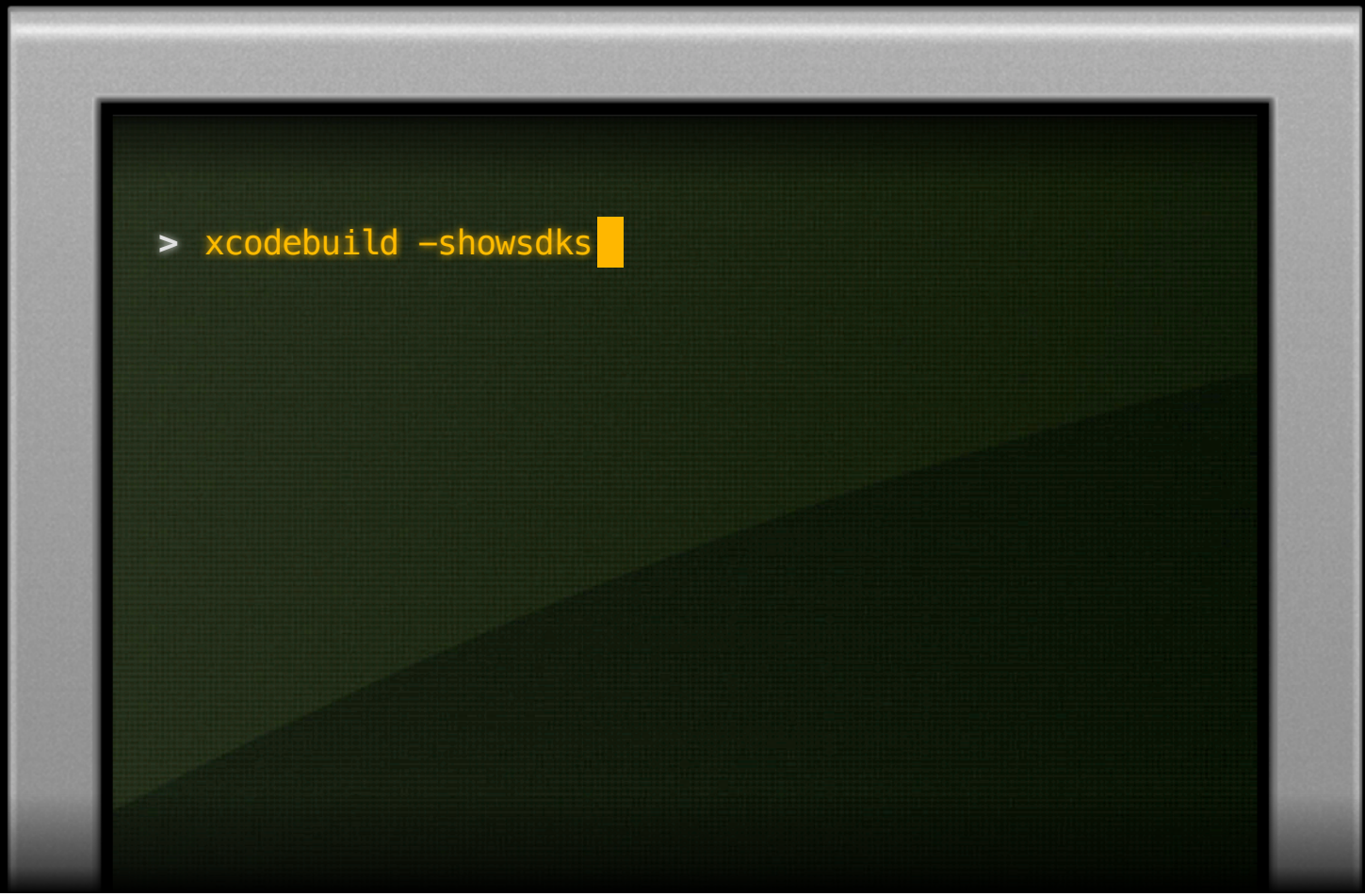
Specifying a Scheme

Specifying a Scheme



Listing the SDKs

Listing the SDKs

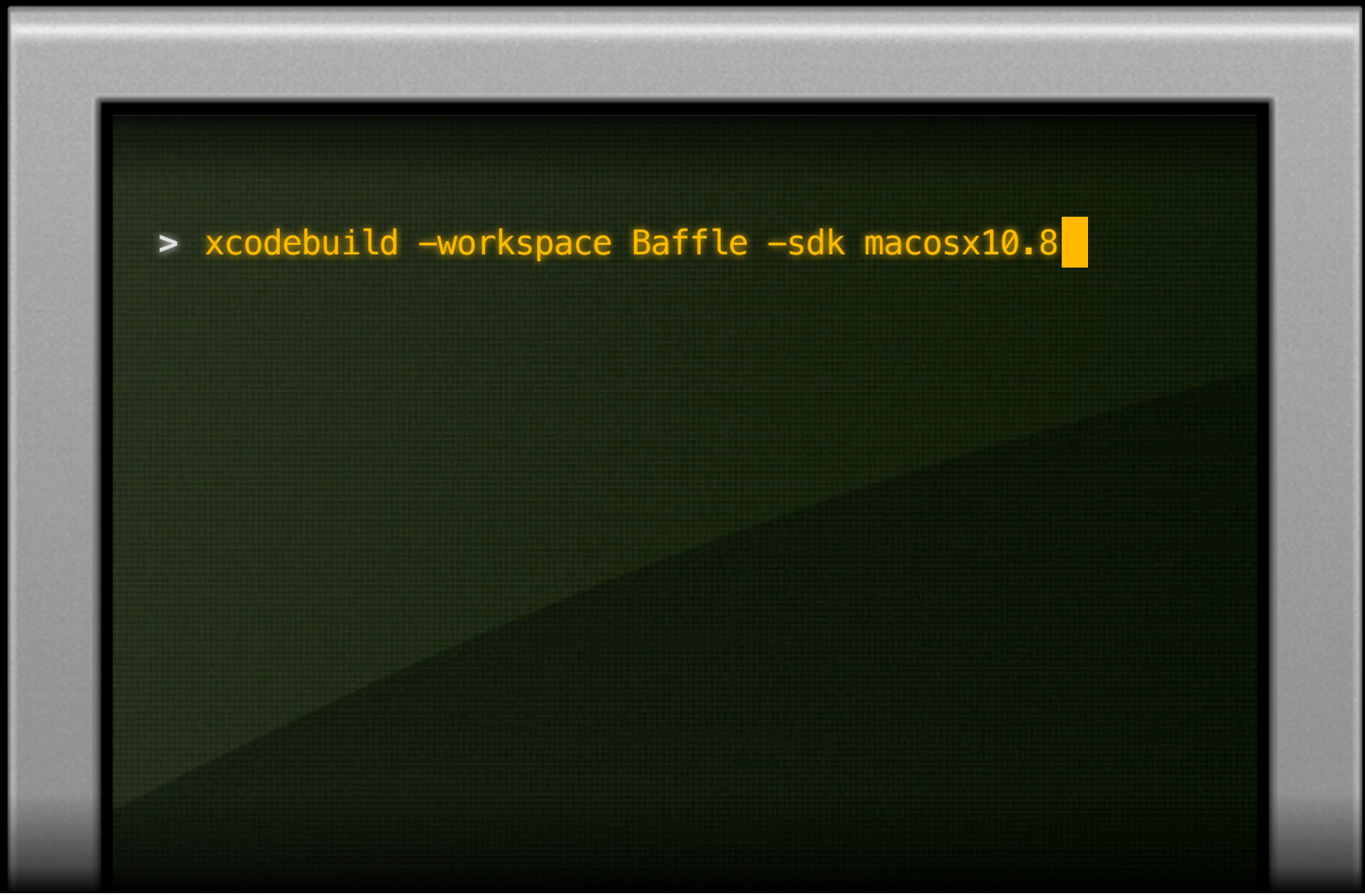


Listing the SDKs

```
> xcodebuild -showsdks

OS X SDKs:
    Mac OS X 10.7               -sdk macosx10.7
    OS X 10.8                   -sdk macosx10.8

iOS SDKs:
    iOS 6.0                     -sdk iphoneos6.0

iOS Simulator SDKs:
    Simulator - iOS 6.0         -sdk iphonesimulator6.0


>
```

Specifying an SDK

Specifying an SDK



Build Settings

Build Settings

- “Knobs and switches” of the build system

Build Settings

- “Knobs and switches” of the build system
- View and edit in Build Settings editor in IDE

Build Settings

- "Knobs and switches" of the build system
- View and edit in Build Settings editor in IDE
- Override by passing on `xcodebuild` command line

Build Settings

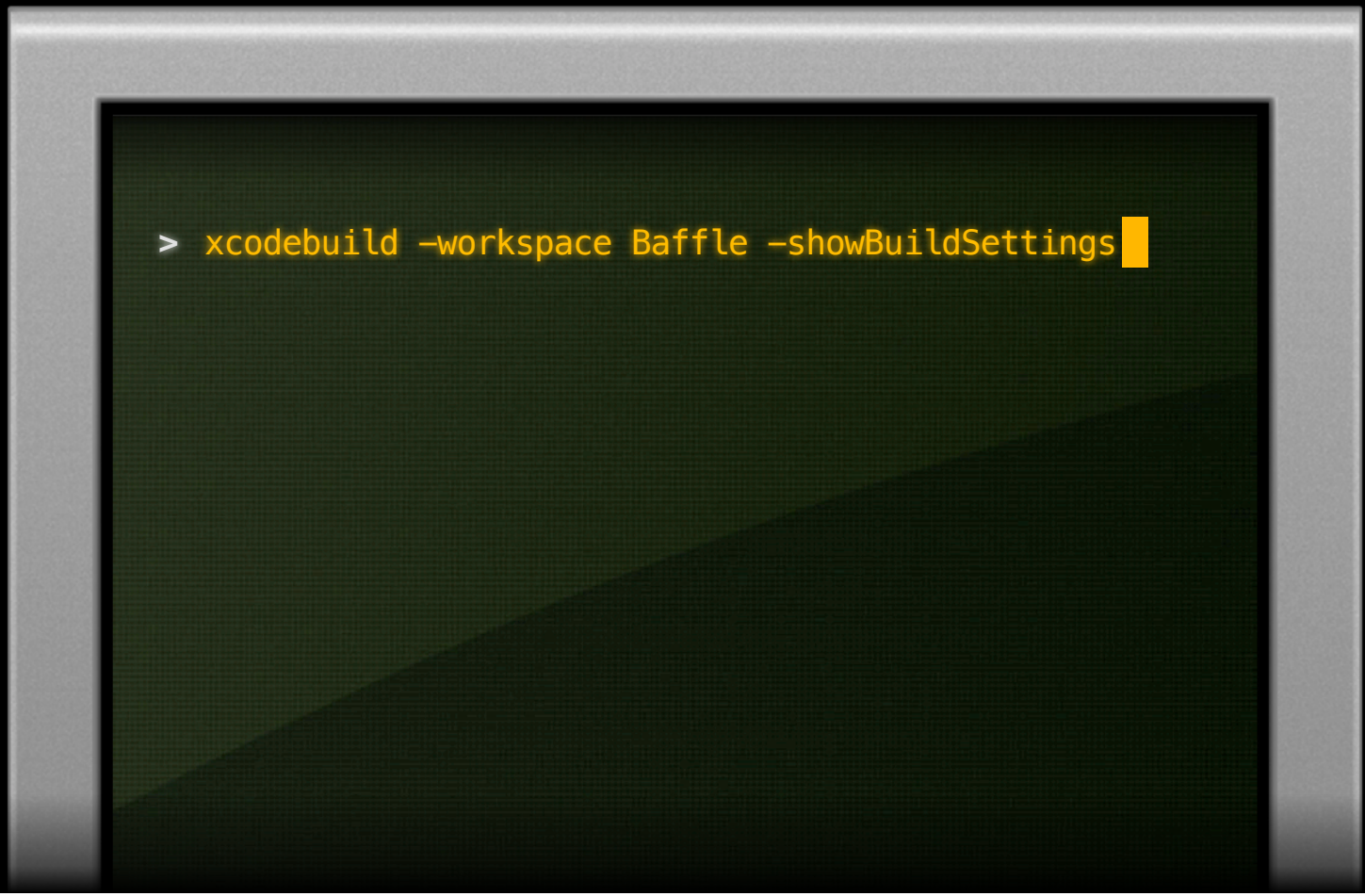
- "Knobs and switches" of the build system
- View and edit in Build Settings editor in IDE
- Override by passing on `xcodebuild` command line
- Set in environment to provide defaults

Build Settings

- "Knobs and switches" of the build system
- View and edit in Build Settings editor in IDE
- Override by passing on `xcodebuild` command line
- Set in environment to provide defaults
- Use `-showBuildSettings` option to `xcodebuild`

Seeing Values of Build Settings

Seeing Values of Build Settings

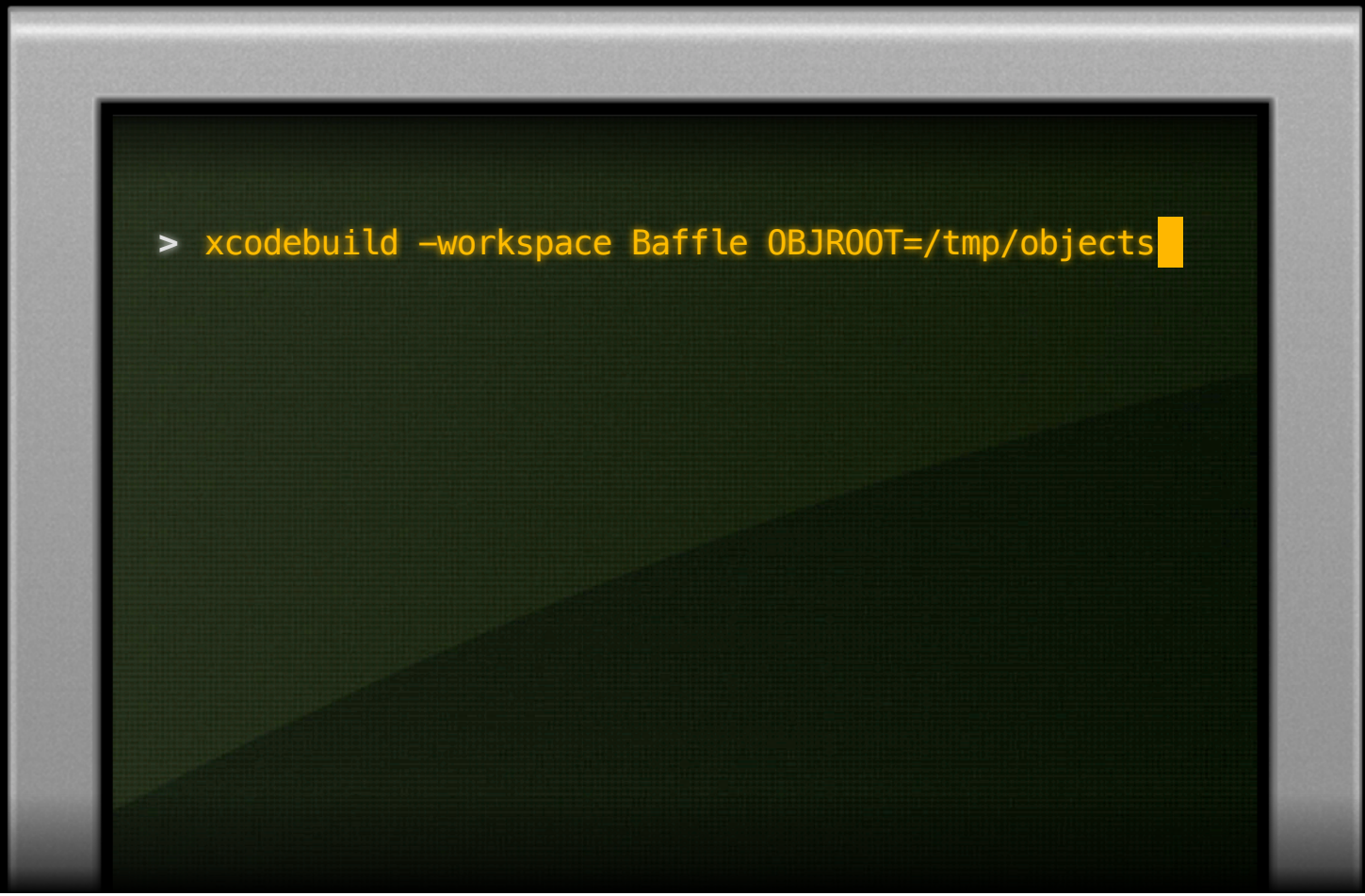


Seeing Values of Build Settings

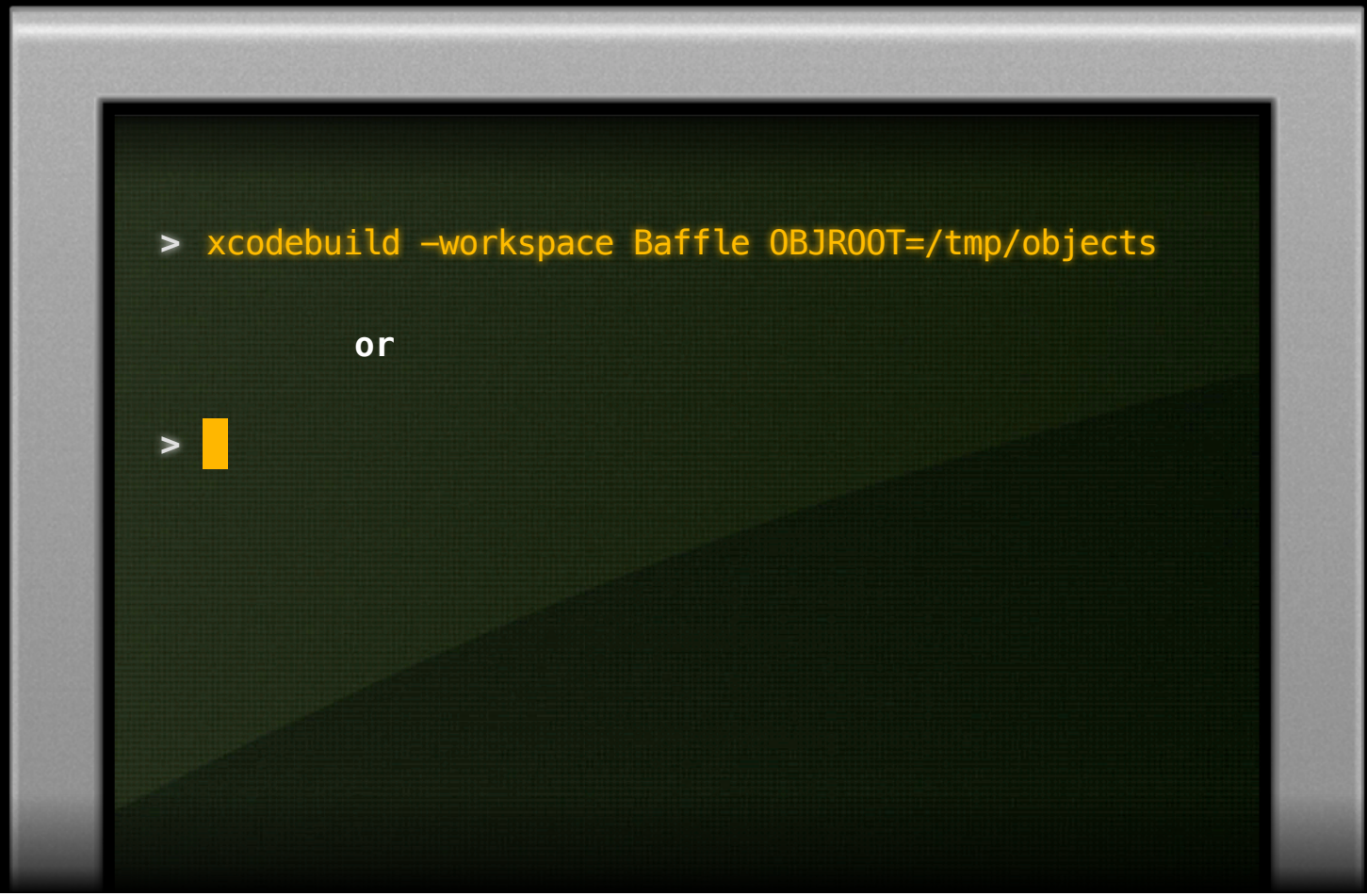
```
> xcodebuild -workspace Baffle -showBuildSettings

Build settings for action build & target "Baffle OSX"
    ACTION = build
    ALWAYS_SEARCH_USER_PATHS = NO
    APPLY_RULES_IN_COPY_FILES = NO
    ARCHS = i386
    ARCHS_STANDARD_32_64_BIT = x86_64 i386
    ARCHS_STANDARD_32_BIT = i386
    ARCHS_STANDARD_64_BIT = x86_64
    AVAILABLE_PLATFORMS = macosx iphoneos
    BUILD_DIR = ./build/Baffle/Build/Products
    CONFIGURATION = Debug
    COPYING_PRESERVES_HFS_DATA = NO

                          :
```

Overriding Build Settings

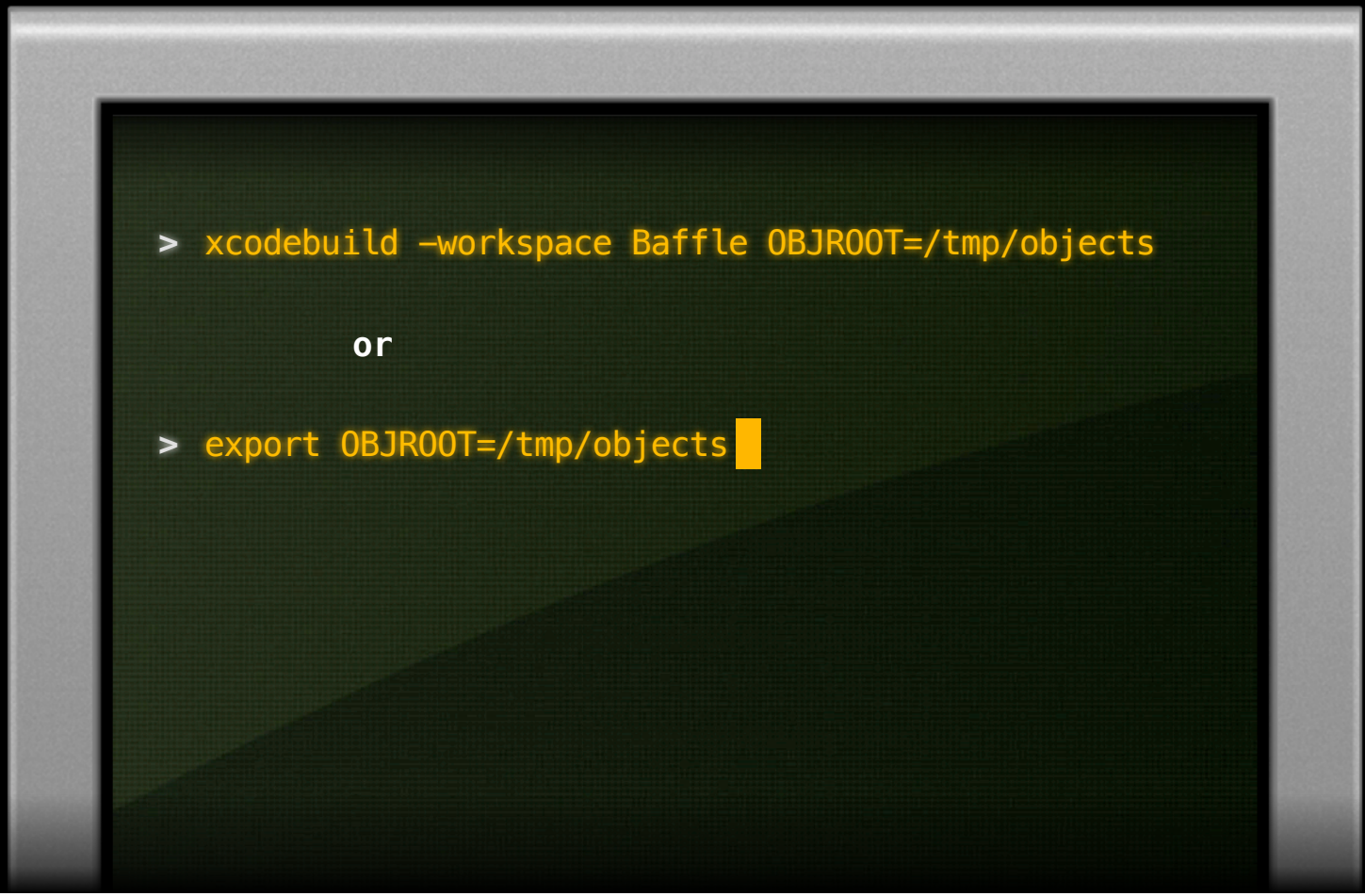
Overriding Build Settings



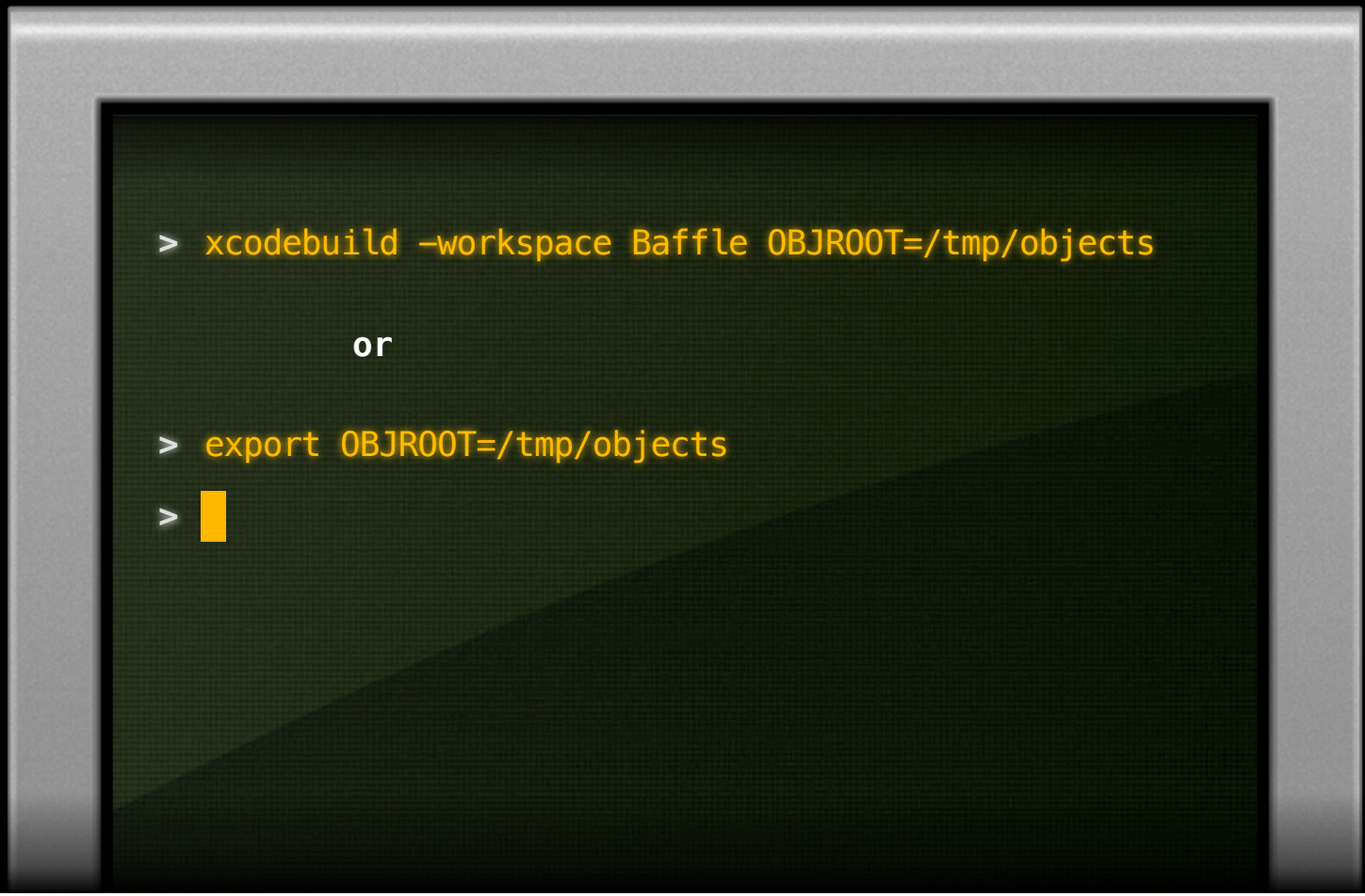
Overriding Build Settings



Overriding Build Settings



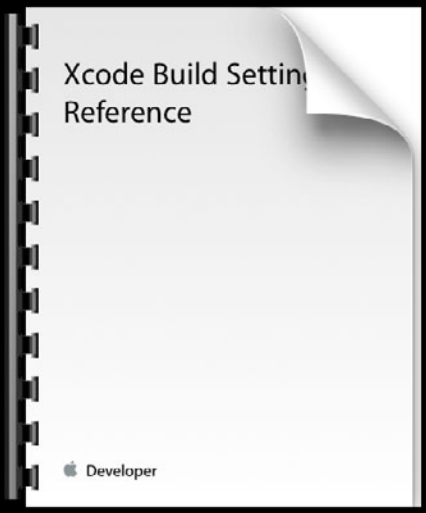
Overriding Build Settings



Overriding Build Settings

```
> xcodebuild -workspace Baffle OBJROOT=/tmp/objects

        or

> export OBJROOT=/tmp/objects
> xcodebuild -workspace Baffle
```

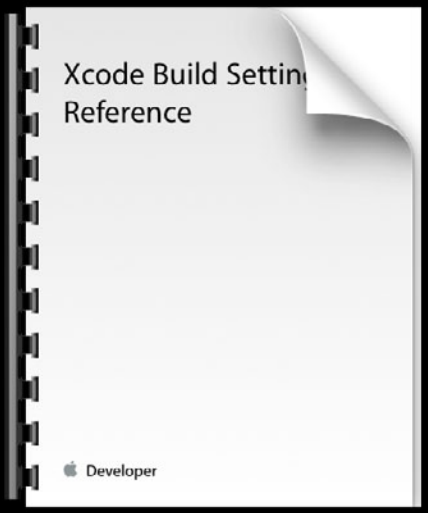
What Build Settings Are Available?



Xcode Build Setting Reference

What Build Settings Are Available?

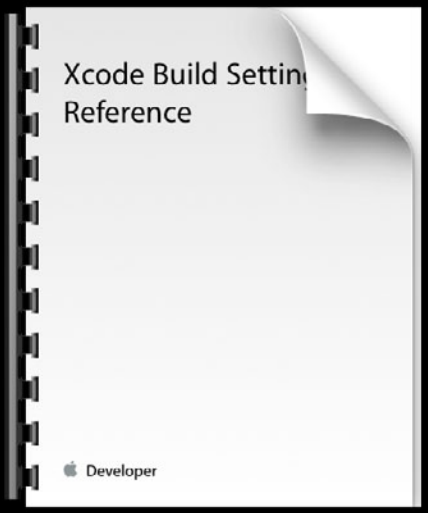
- Xcode built-in documentation



Xcode Build Setting Reference

What Build Settings Are Available?

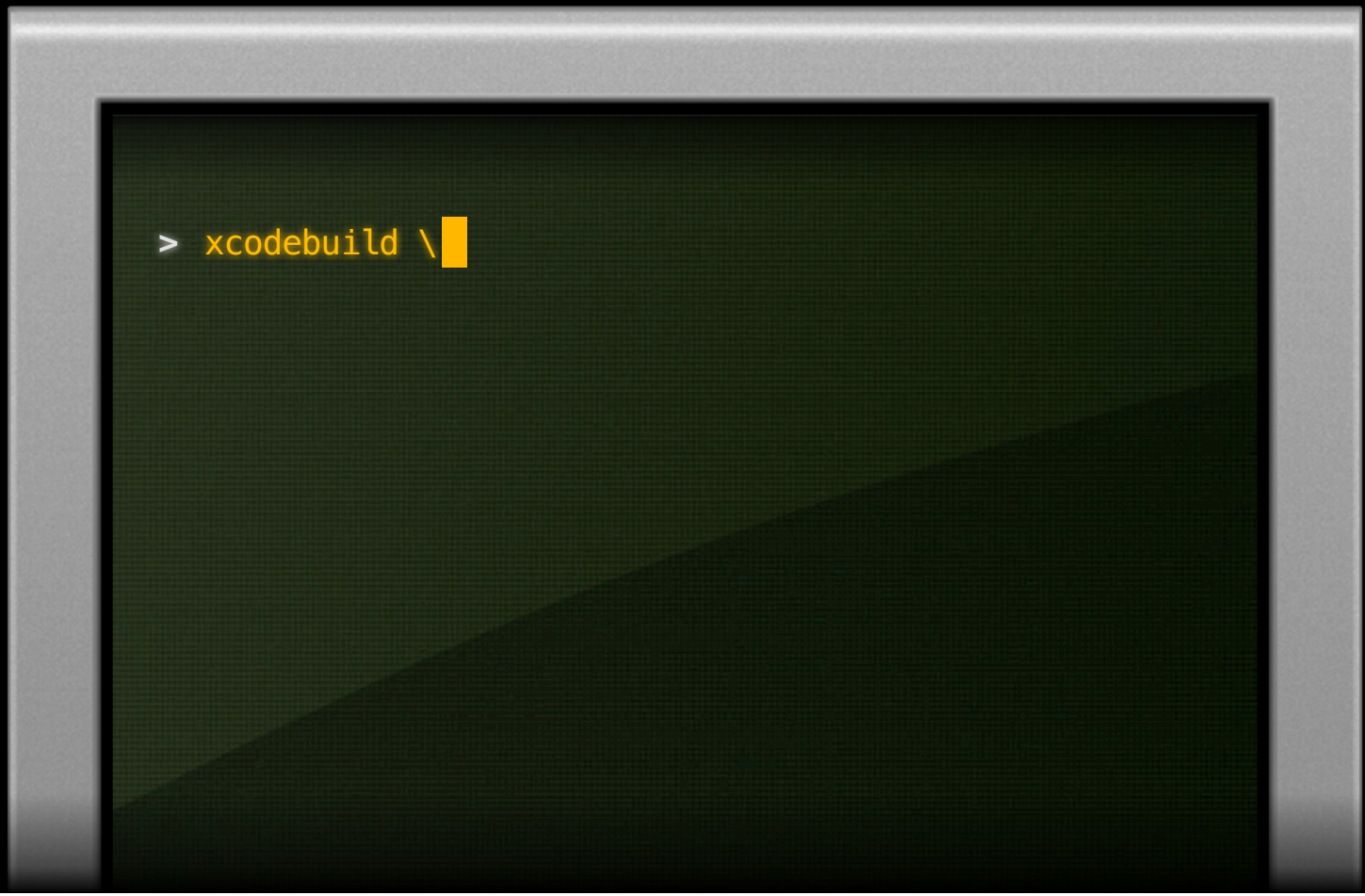
- Xcode built-in documentation
- Developer resources



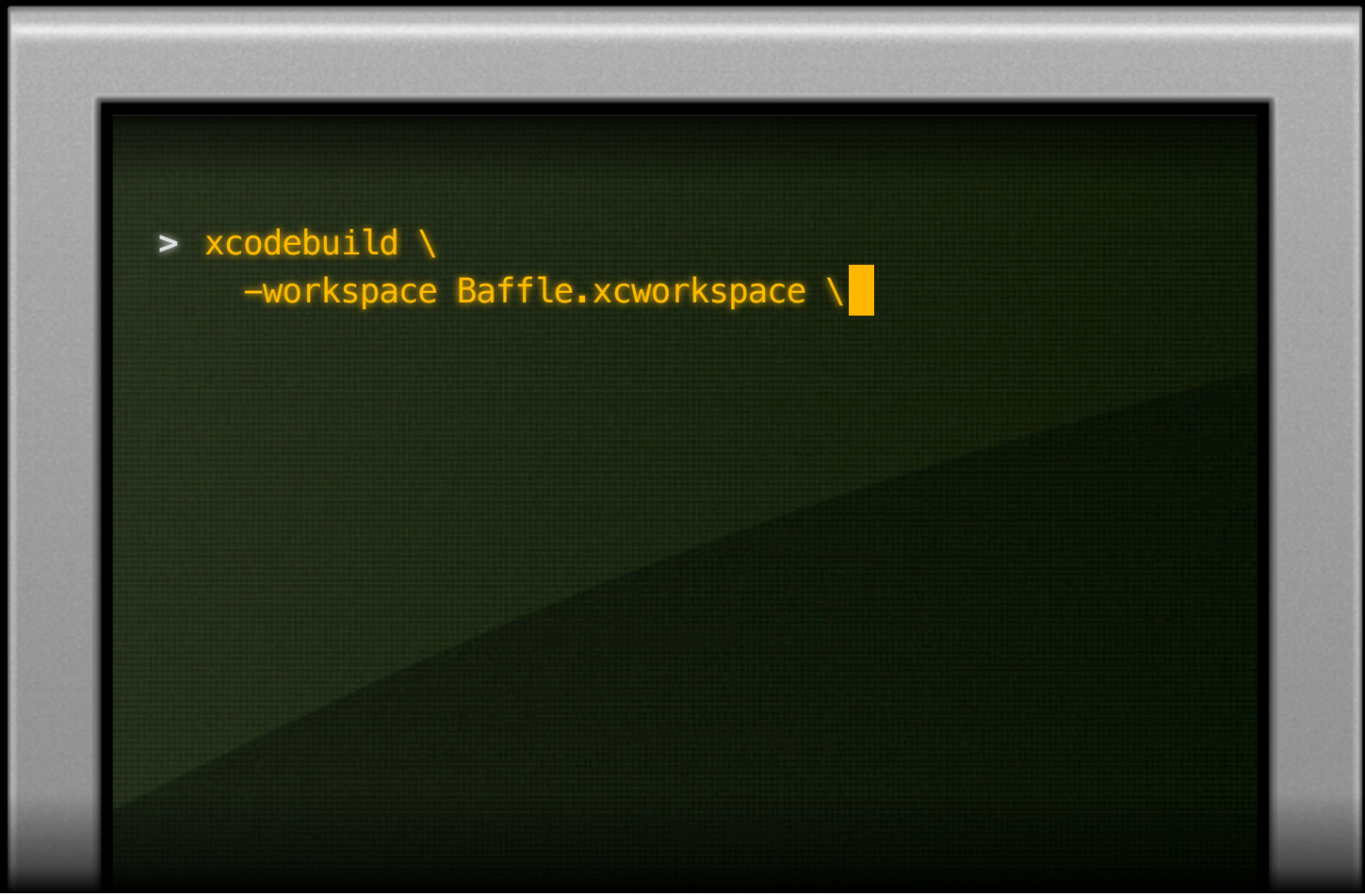
Xcode Build Setting Reference

Putting It All Together

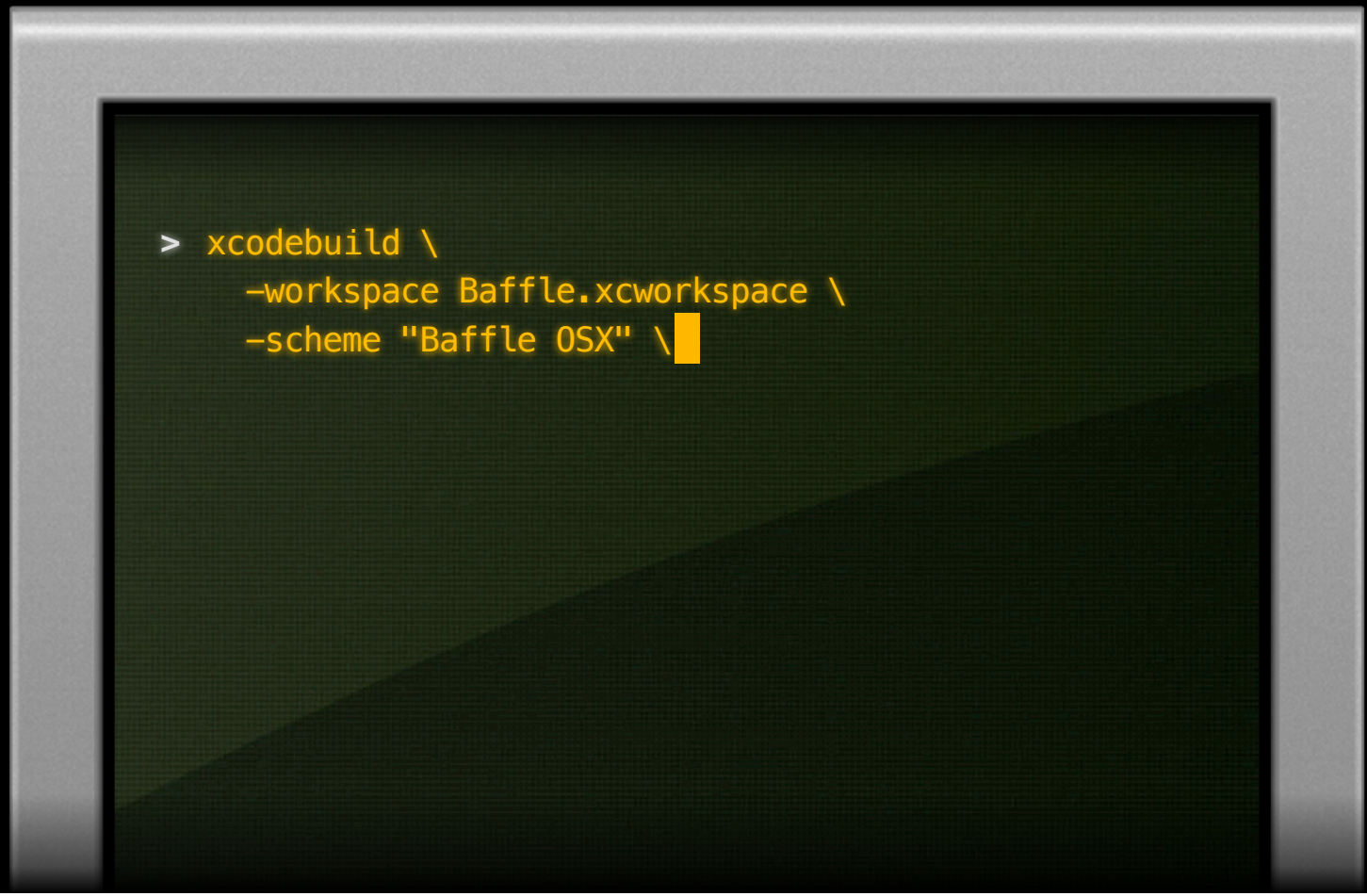
Putting It All Together



Putting It All Together



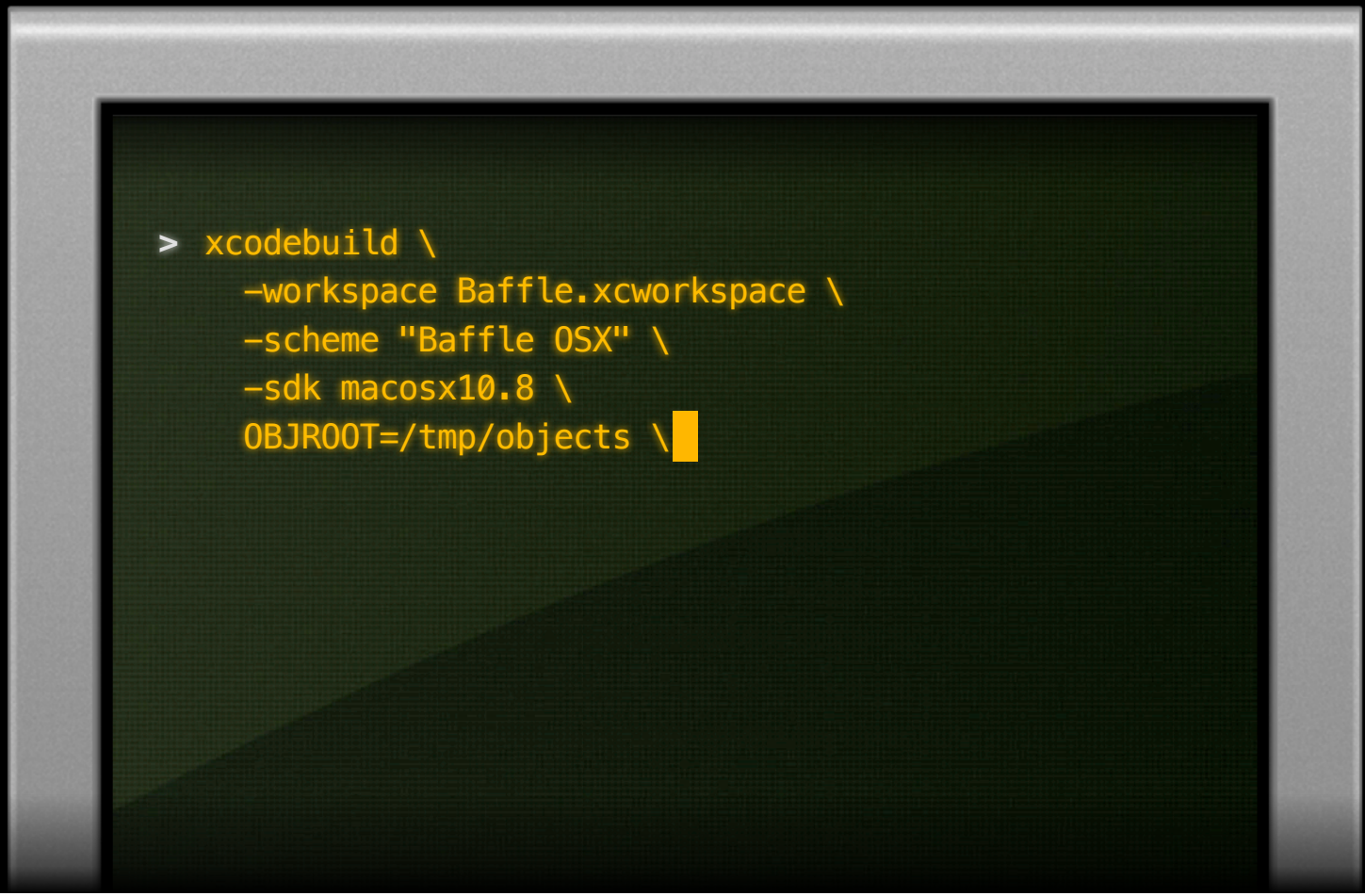
Putting It All Together



Putting It All Together

```
> xcodebuild \
    -workspace Baffle.xcworkspace \
    -scheme "Baffle OSX" \
    -sdk macosx10.8 \
```

Putting It All Together



Putting It All Together

```
> xcodebuild \
    -workspace Baffle.xcworkspace \
    -scheme "Baffle OSX" \
    -sdk macosx10.8 \
    OBJROOT=/tmp/objects \
    SYMROOT=/tmp/symbols \
```

Putting It All Together

```
> xcodebuild \
    -workspace Baffle.xcworkspace \
    -scheme "Baffle OSX" \
    -sdk macosx10.8 \
    OBJROOT=/tmp/objects \
    SYMROOT=/tmp/symbols \
    DSTROOT=/tmp/distrib
```

Which Xcode.app?

Xcode 4.3

Which Xcode.app?

Xcode 4.3

Xcode 4.5 Preview

Which Xcode.app?

Xcode 4.3

Xcode 4.5 Preview

Which Xcode.app?

Xcode 4.3

Xcode 4.5 Preview

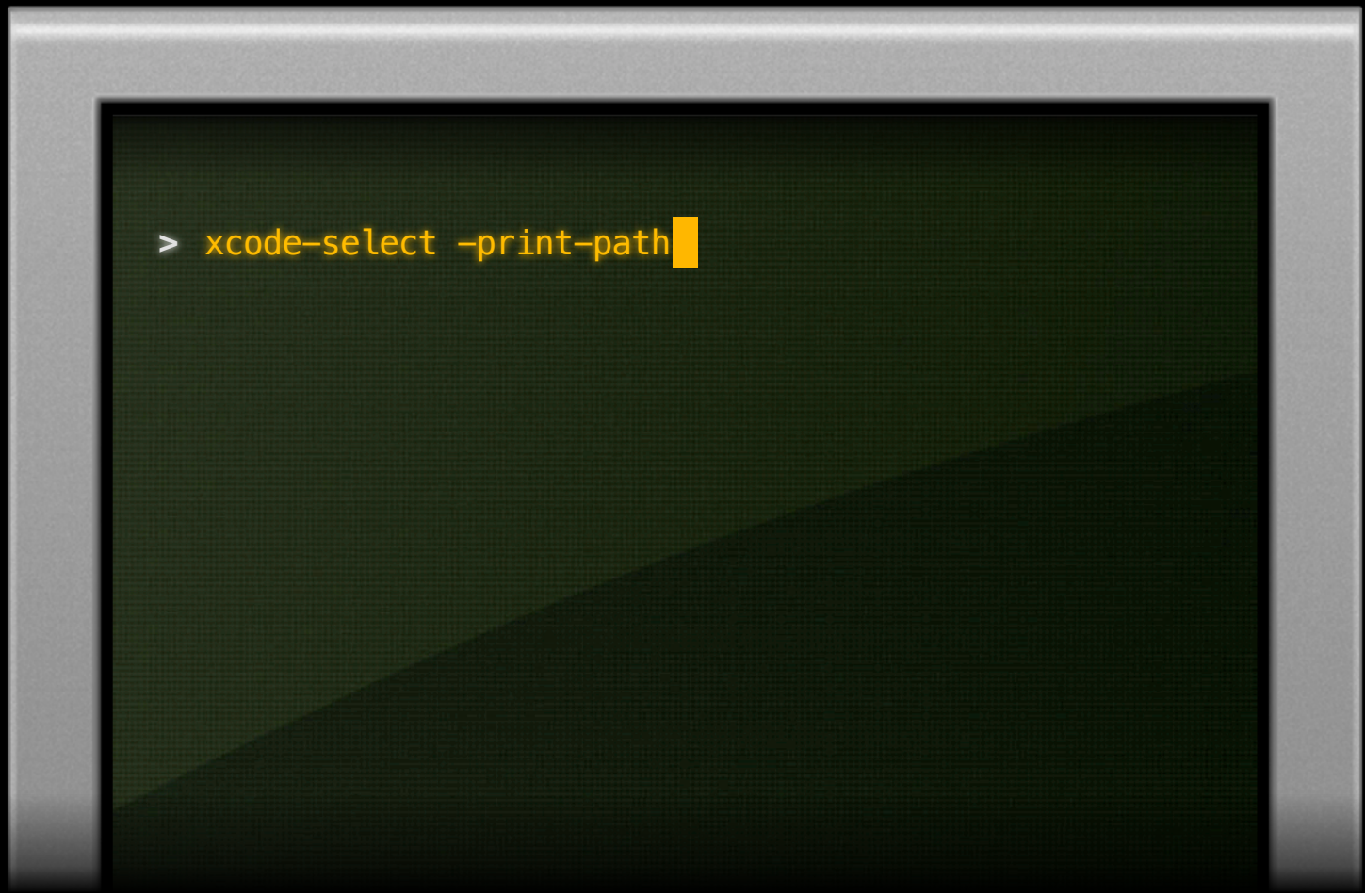
Which Xcode.app?

Xcode 4.3

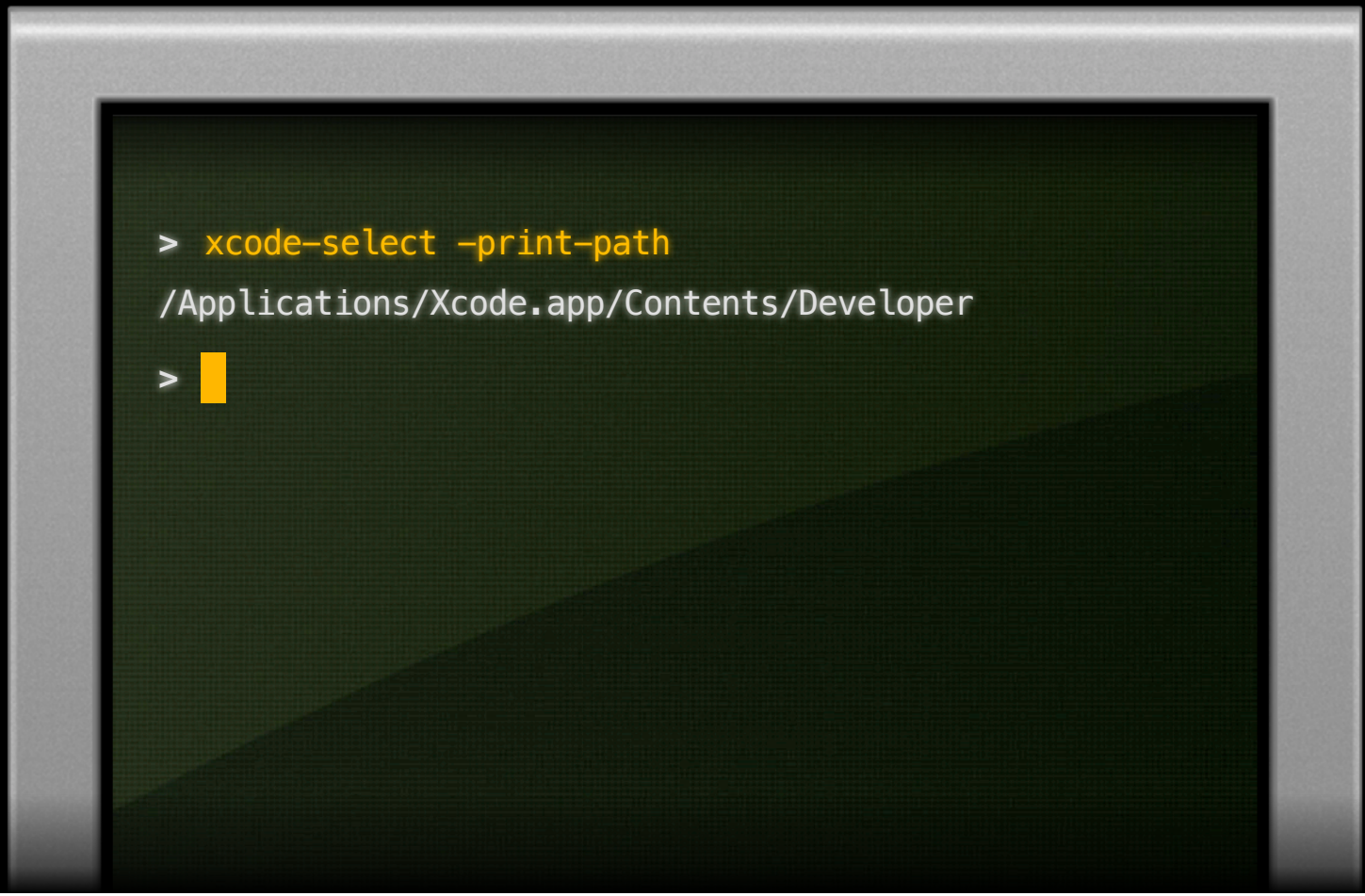
Xcode 4.5 Preview

Use `xcode-select`

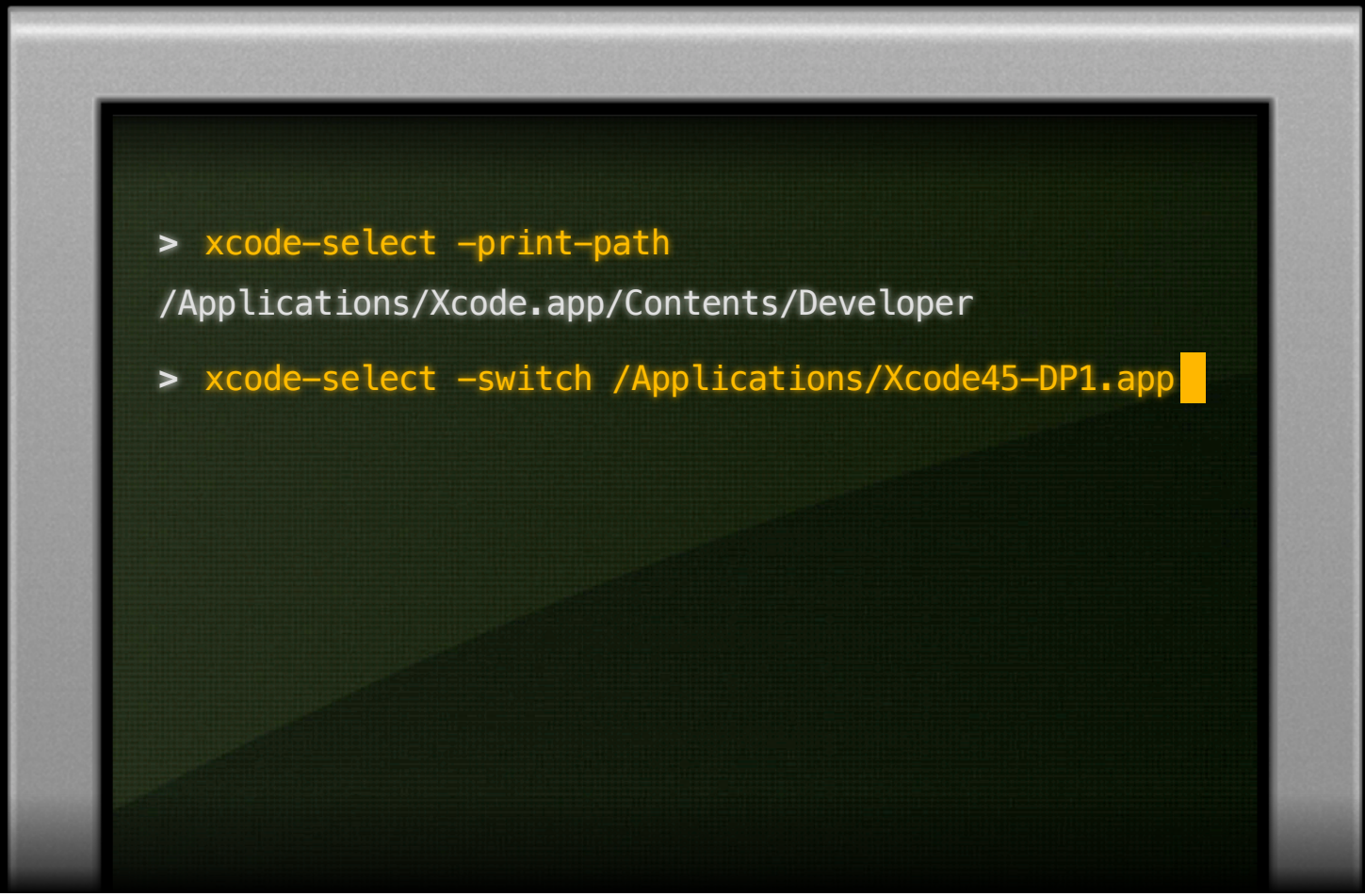
Use xcode-select



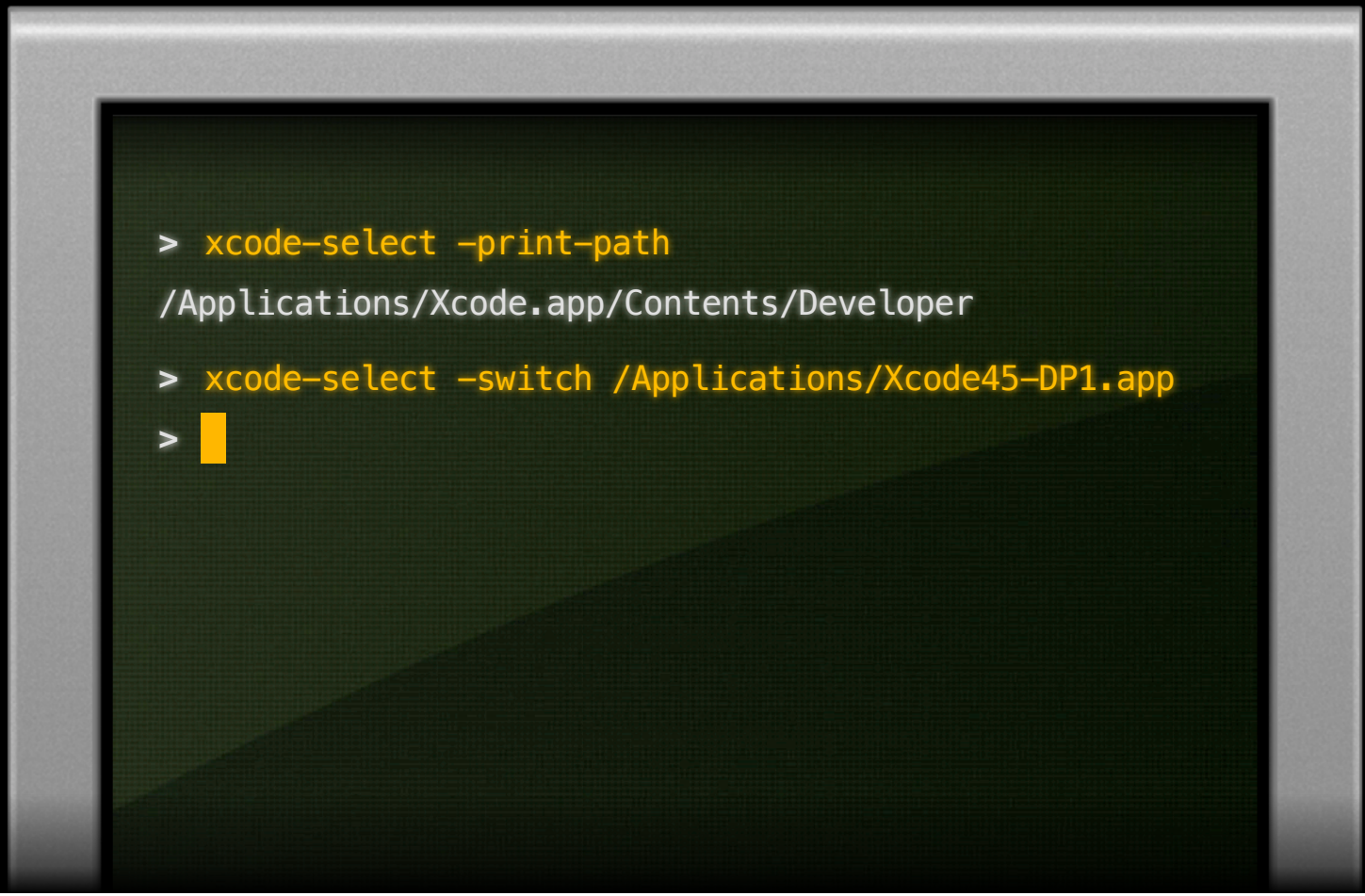
Use xcode-select



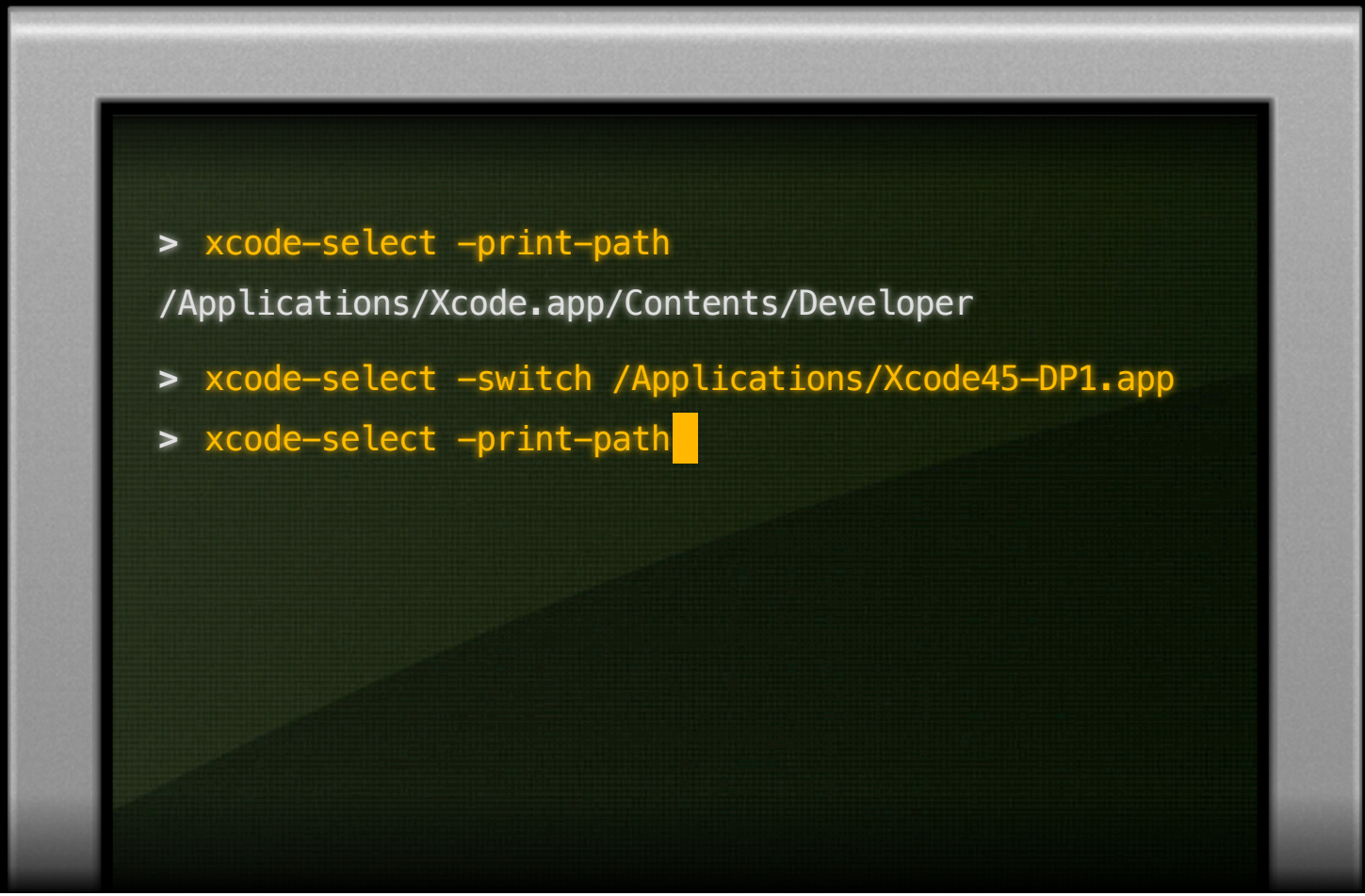
Use xcode-select



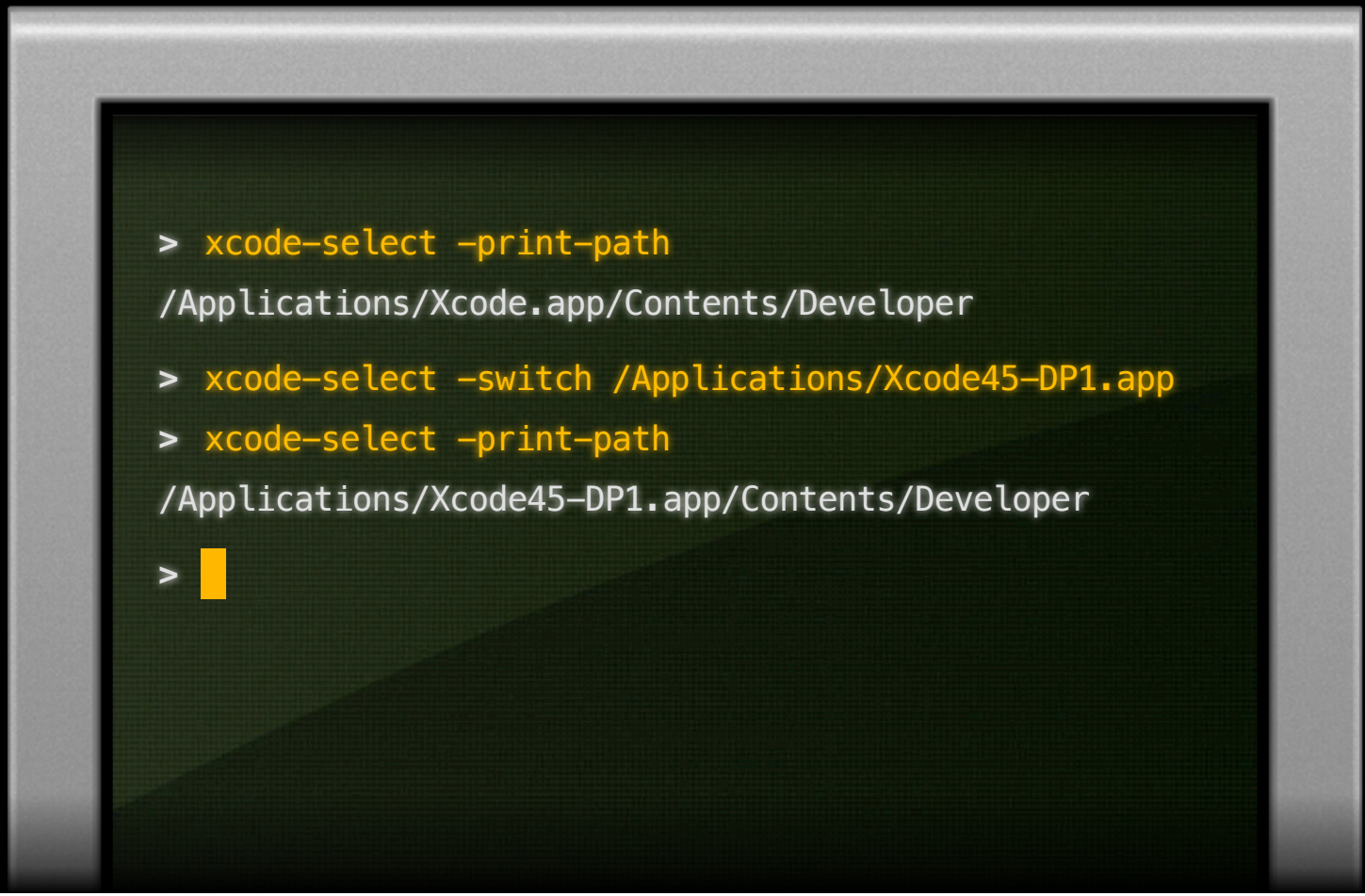
Use xcode-select



Use xcode-select



Use xcode-select



Use xcode-select

```
> xcode-select -print-path
/Applications/Xcode.app/Contents/Developer
> xcode-select -switch /Applications/Xcode45-DP1.app
> xcode-select -print-path
/Applications/Xcode45-DP1.app/Contents/Developer
> xcodebuild -version
```

Use xcode-select

```
> xcode-select -print-path
/Applications/Xcode.app/Contents/Developer
> xcode-select -switch /Applications/Xcode45-DP1.app
> xcode-select -print-path
/Applications/Xcode45-DP1.app/Contents/Developer
> xcodebuild -version
Xcode 4.5
Build version 4G78z
>
```

What Is `xcode-select` Really Doing?

- Sets the Xcode.app that's used by tools in /usr/bin

What Is `xcode-select` Really Doing?

- Sets the Xcode.app that's used by tools in /usr/bin
  - `xcodebuild`

What Is `xcode-select` Really Doing?

- Sets the Xcode.app that's used by tools in /usr/bin
  - `xcodebuild`
  - `xcrun`

What Is `xcode-select` Really Doing?

- Sets the Xcode.app that's used by tools in /usr/bin
  - `xcodebuild`
  - `xcrun`
  - `opendiff`

What Is `xcode-select` Really Doing?

- Sets the Xcode.app that's used by tools in /usr/bin
  - `xcodebuild`
  - `xcrun`
  - `opendiff`
  - `instruments`

What Is `xcode-select` Really Doing?

- Sets the Xcode.app that's used by tools in /usr/bin
  - `xcodebuild`
  - `xcrun`
  - `opendiff`
  - `instruments`
  - `ibtool`

What Is `xcode-select` Really Doing?

- Sets the Xcode.app that's used by tools in /usr/bin
  - `xcodebuild`
  - `xcrun`
  - `opendiff`
  - `instruments`
  - `ibtool`
  - `xed`

What Is `xcode-select` Really Doing?

- Sets the Xcode.app that's used by tools in /usr/bin
  - `xcodebuild`
  - `xcrun`
  - `opendiff`
  - `instruments`
  - `ibtool`
  - `xed`
  - `iprofiler`

What Is `xcode-select` Really Doing?

- Sets the Xcode.app that's used by tools in /usr/bin
  - `xcodebuild`
  - `xcrun`
  - `opendiff`
  - `instruments`
  - `ibtool`
  - `xed`
  - `iprofiler`
  - `agvtool`

What Is `xcode-select` Really Doing?

- Sets the Xcode.app that's used by tools in /usr/bin
  - `xcodebuild`
  - `xcrun`
  - `opendiff`
  - `instruments`
  - `ibtool`
  - `xed`
  - `iprofiler`
  - `agvtool`

What Is `xcrun`?

What Is `xcrun`?

- Find or run a tool inside Xcode.app

What Is `xcrun`?

- Find or run a tool inside Xcode.app
  - Context-sensitive—depends on SDK, etc.

What Is `xcrun`?

- Find or run a tool inside Xcode.app
  - Context-sensitive—depends on SDK, etc.
  - Also searches standard `$PATH` as a fallback

What Is `xcrun`?

- Find or run a tool inside Xcode.app
  - Context-sensitive—depends on SDK, etc.
  - Also searches standard `$PATH` as a fallback
- Mode 1: Print the path of the tool

What Is `xcrun`?

- Find or run a tool inside Xcode.app
  - Context-sensitive—depends on SDK, etc.
  - Also searches standard `$PATH` as a fallback
- Mode 1: Print the path of the tool
  - Usage: `xcrun -find tool`

What Is `xcrun`?

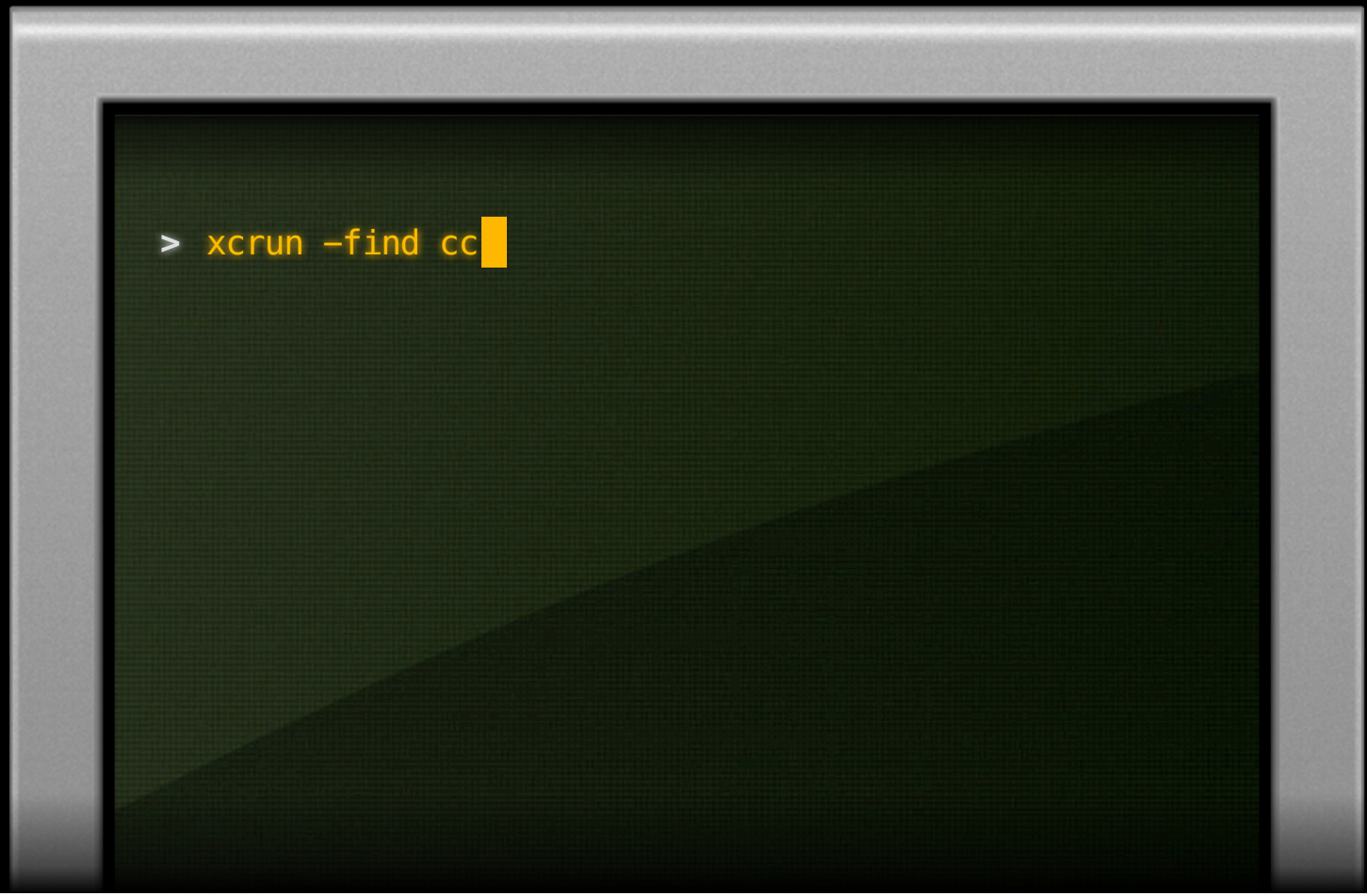
- Find or run a tool inside Xcode.app
  - Context-sensitive—depends on SDK, etc.
  - Also searches standard `$PATH` as a fallback
- Mode 1: Print the path of the tool
  - Usage: `xcrun -find tool`
- Mode 2: Run the tool with a full command line

What Is `xcrun`?

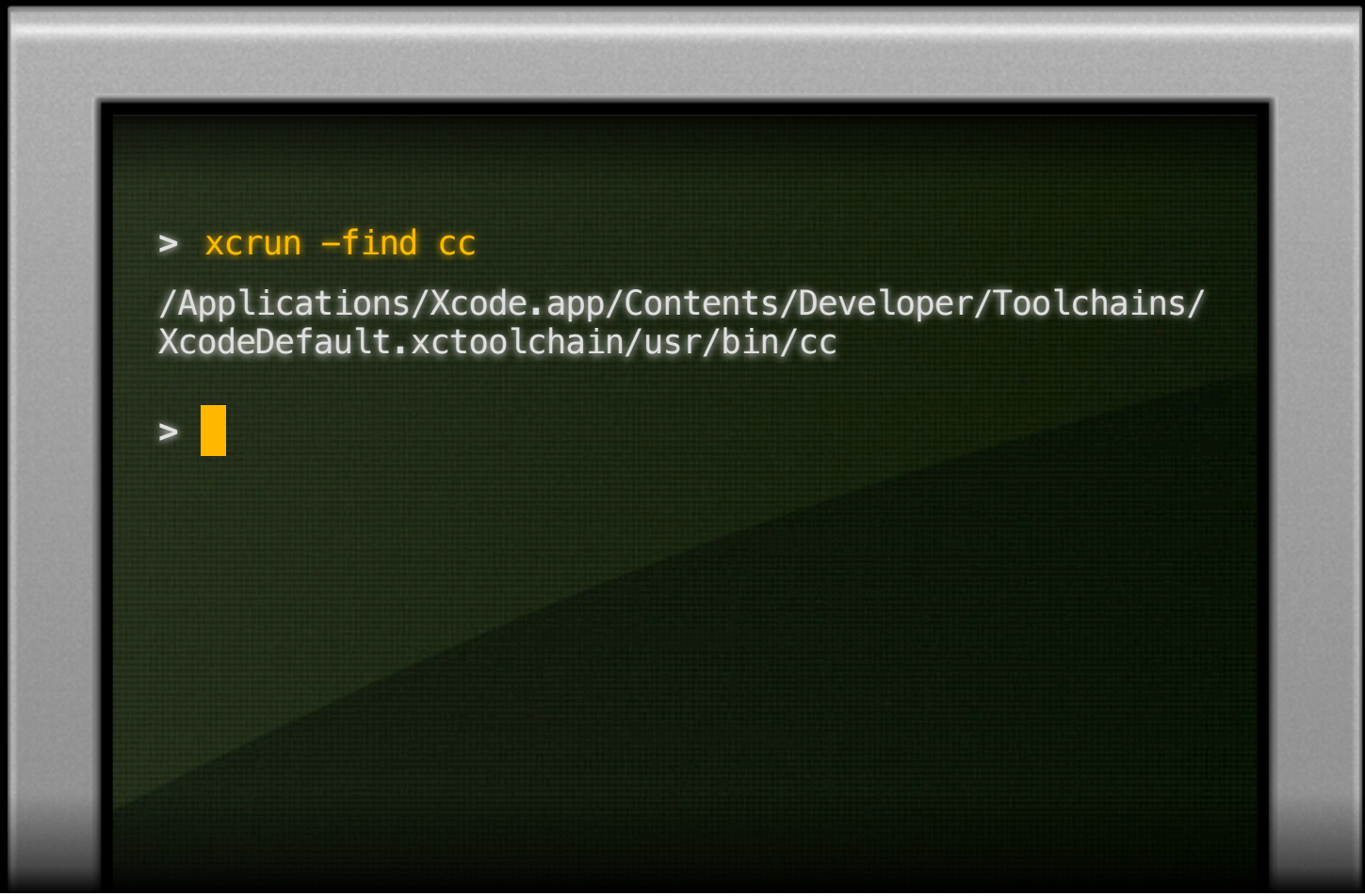
- Find or run a tool inside Xcode.app
  - Context-sensitive—depends on SDK, etc.
  - Also searches standard `$PATH` as a fallback
- Mode 1: Print the path of the tool
  - Usage: `xcrun -find tool`
- Mode 2: Run the tool with a full command line
  - Usage: `xcrun tool arg1 arg2 ...`

Using `xcrun`

Using `xcrun`



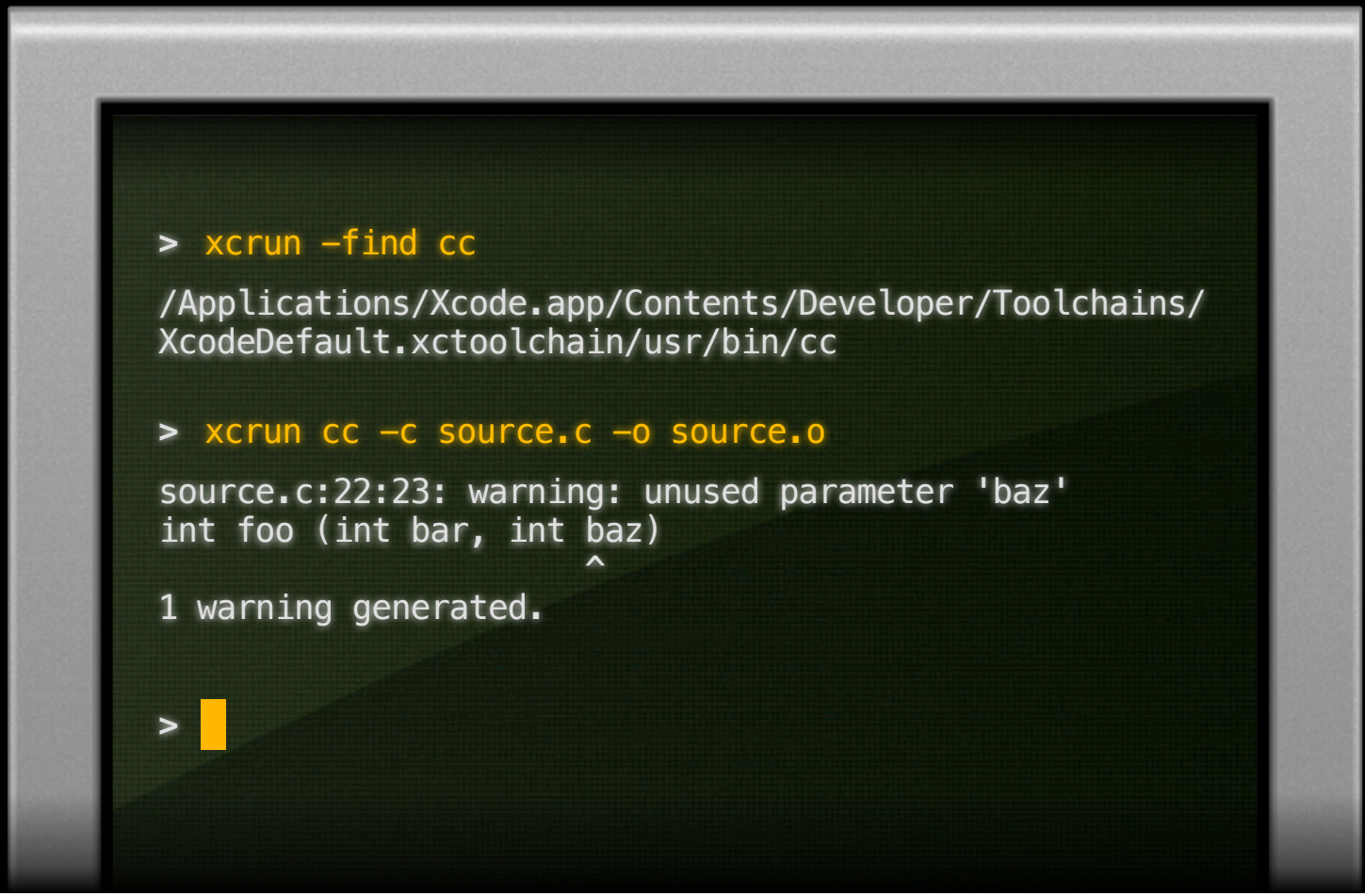
Using `xcrun`


> xcrun -find cc
/Applications/Xcode.app/Contents/Developer/Toolchains/
XcodeDefault.xctoolchain/usr/bin/cc
>

Using `xcrun`

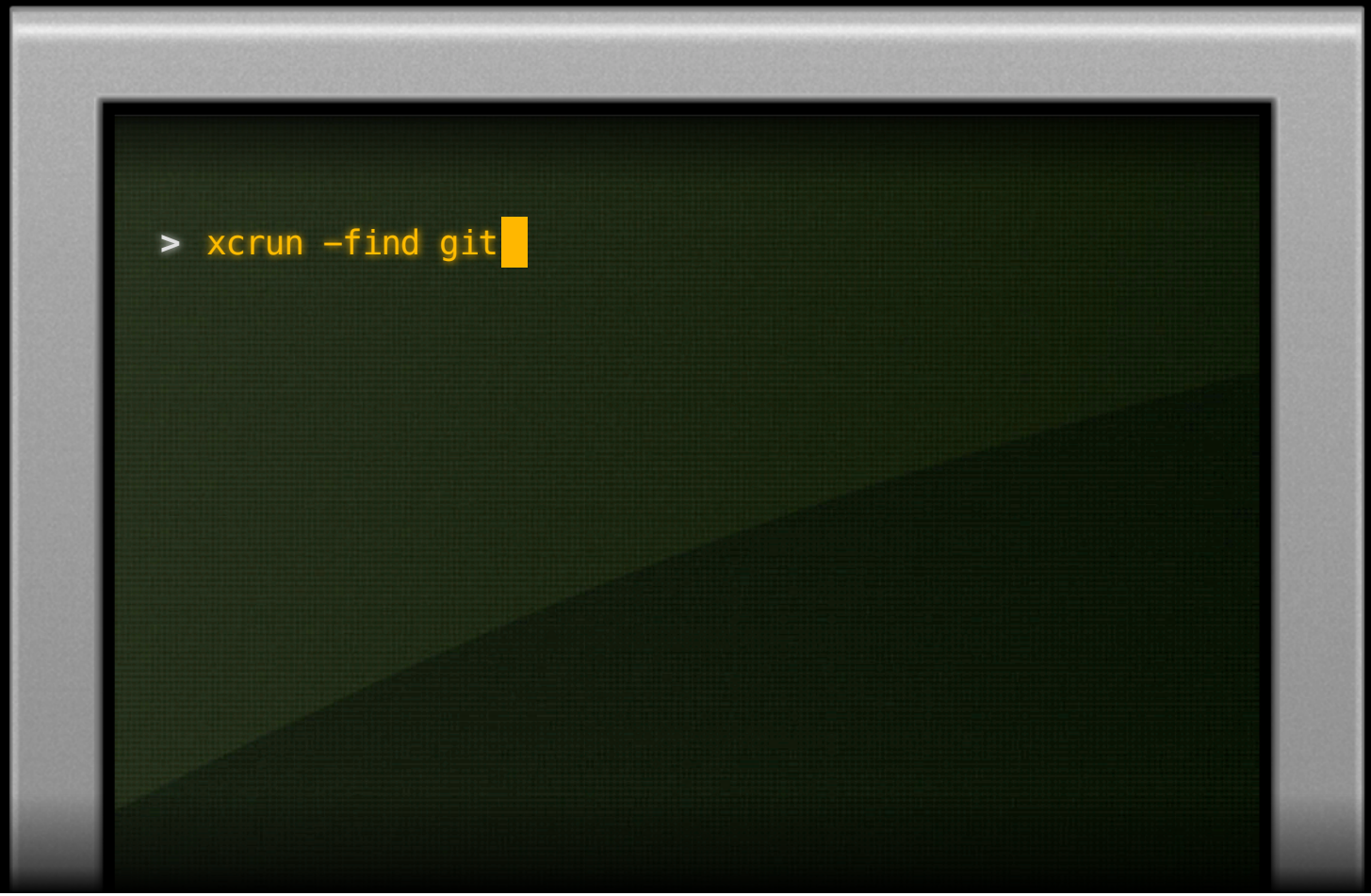
```
> xcrun -find cc
/Applications/Xcode.app/Contents/Developer/Toolchains/
XcodeDefault.xctoolchain/usr/bin/cc

> xcrun cc -c source.c -o source.o
```

Using `xcrun`

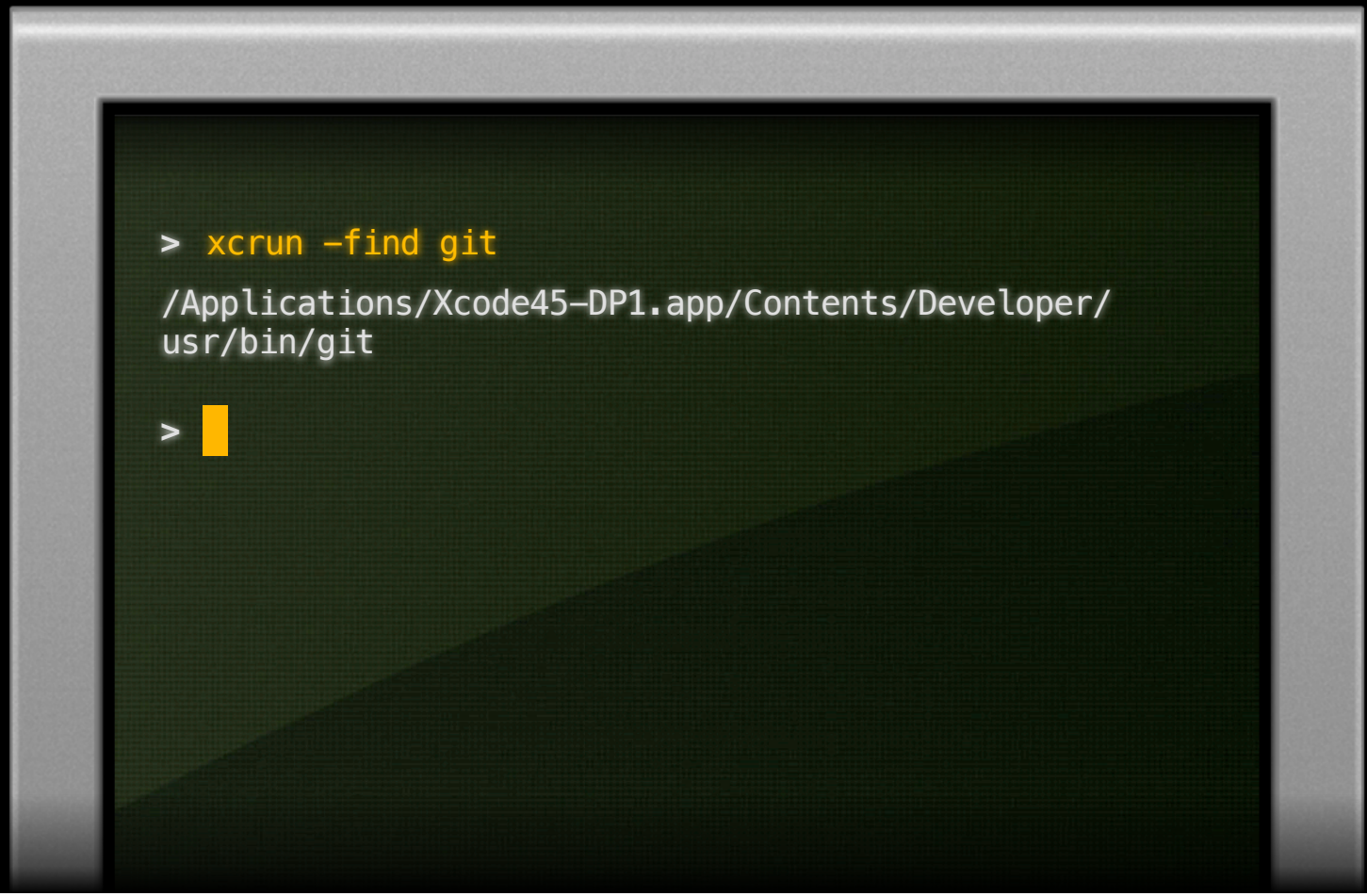

> xcrun -find cc
/Applications/Xcode.app/Contents/Developer/Toolchains/
XcodeDefault.xctoolchain/usr/bin/cc
> xcrun cc -c source.c -o source.o
source.c:22:23: warning: unused parameter 'baz'
int foo (int bar, int baz)
^
1 warning generated.
>

Using `xcrun`

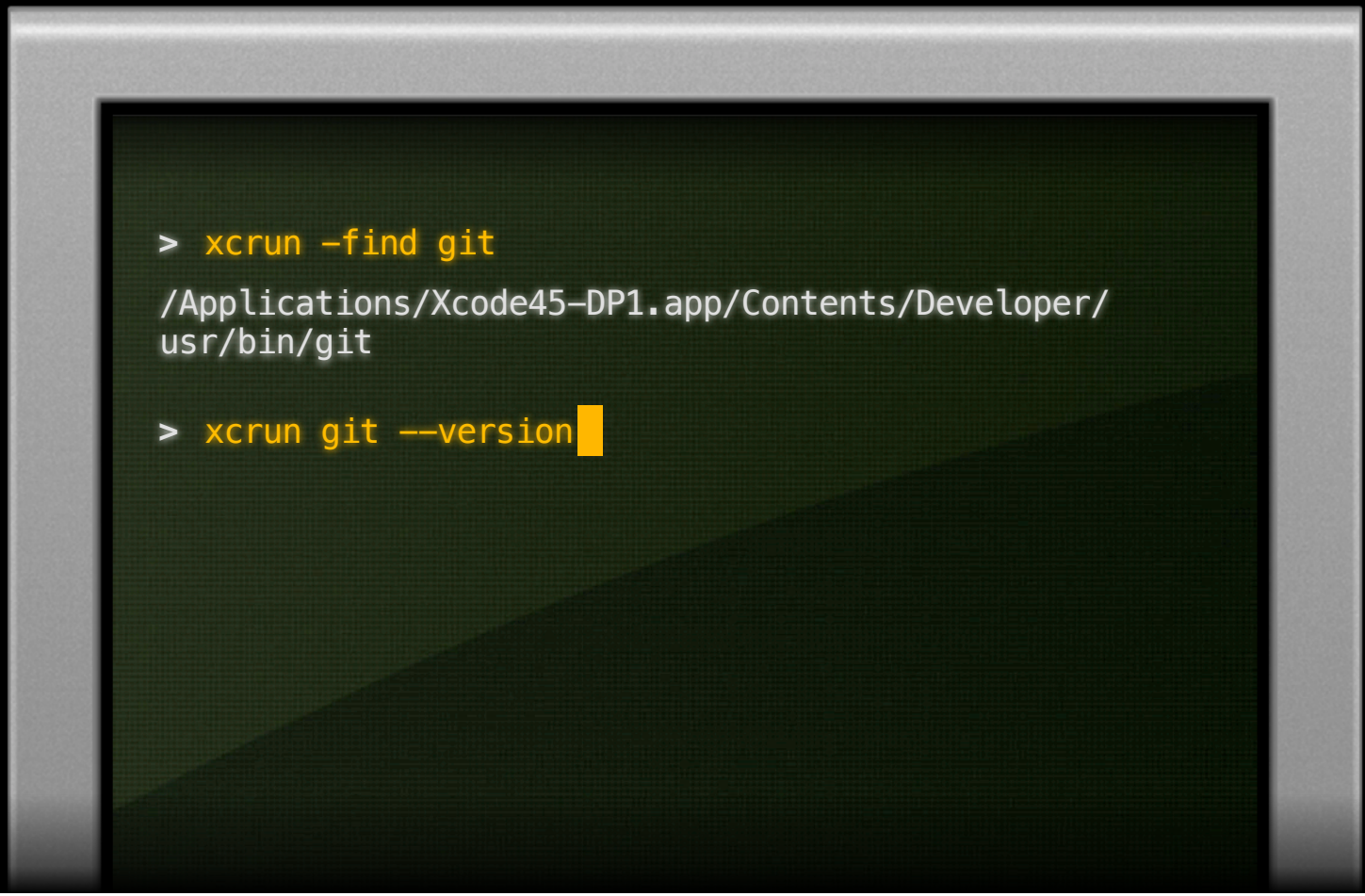
Using `xcrun`



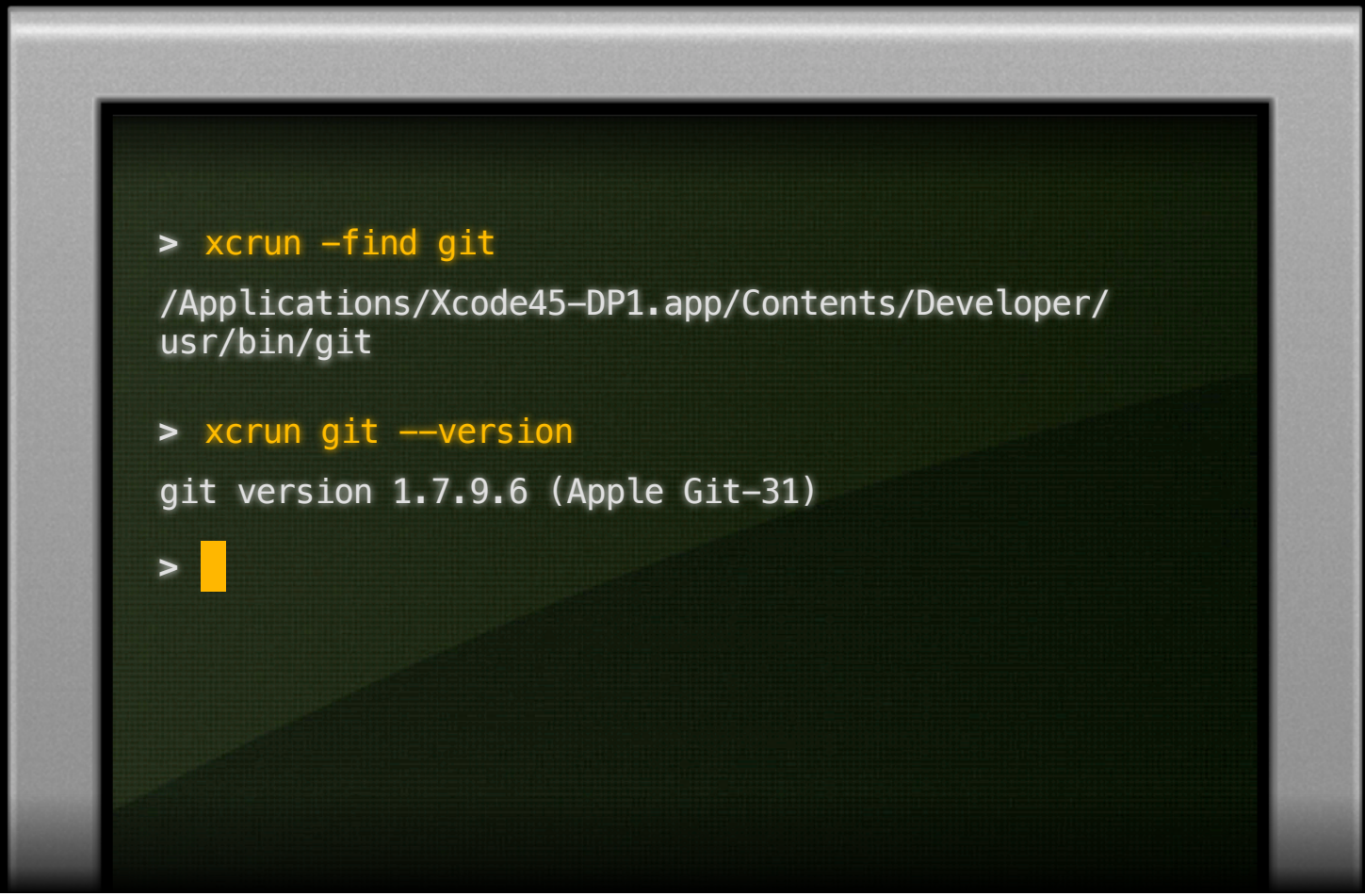
Using `xcrun`


> xcrun -find git
/Applications/Xcode45-DP1.app/Contents/Developer/
usr/bin/git
>

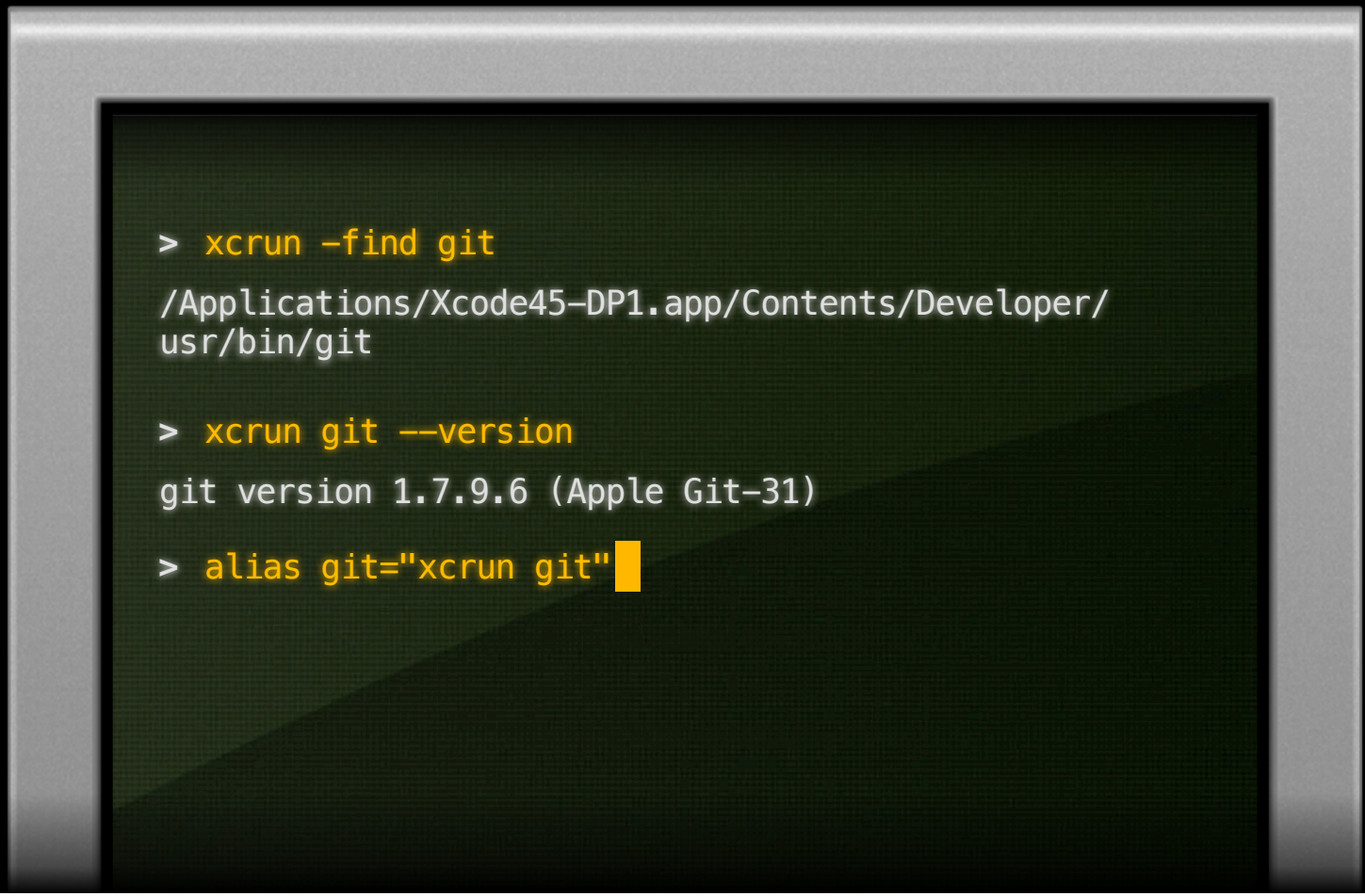
Using `xcrun`


> xcrun -find git
/Applications/Xcode45-DP1.app/Contents/Developer/
usr/bin/git
> xcrun git --version

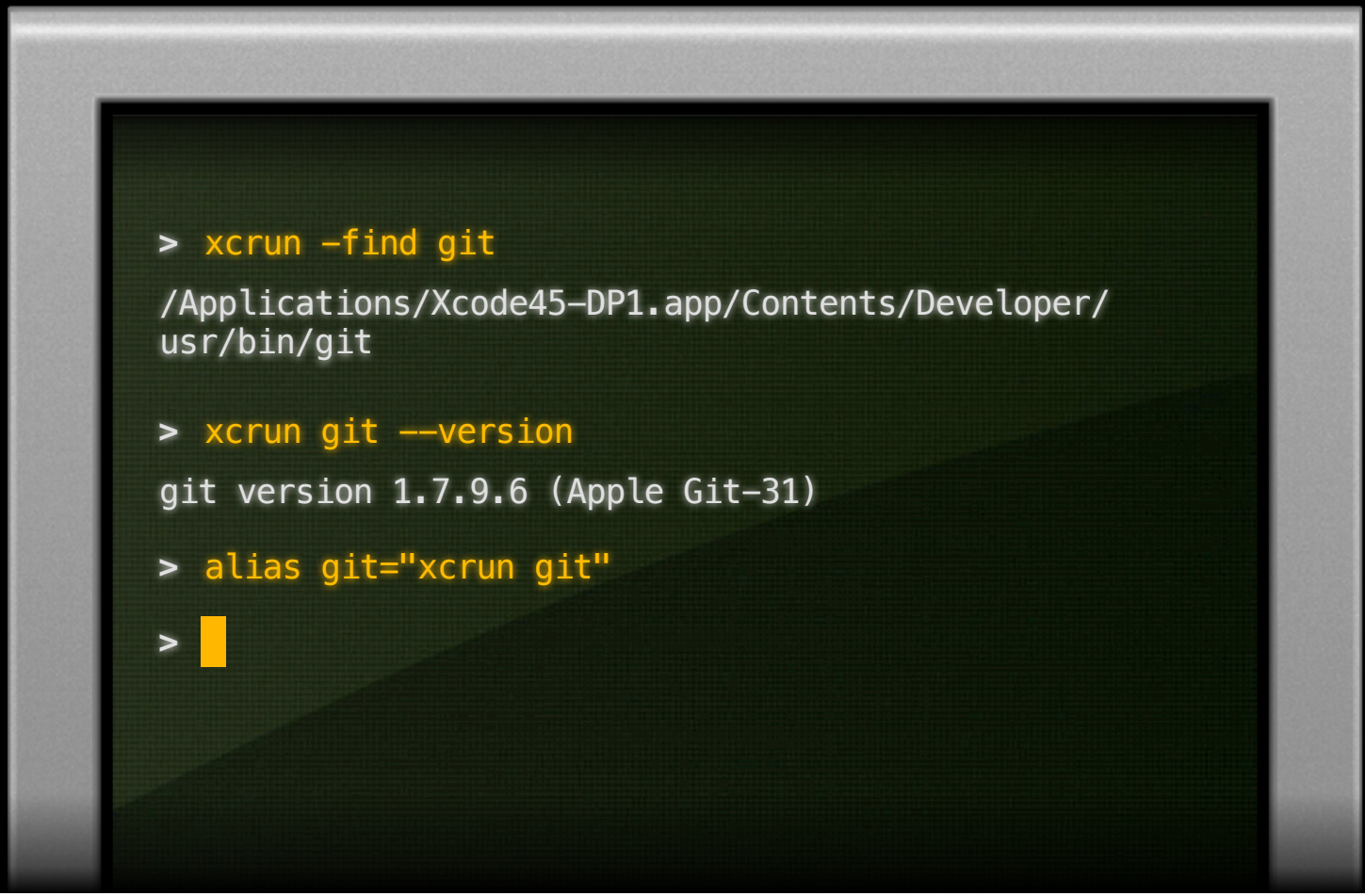
Using `xcrun`


> xcrun -find git
/Applications/Xcode45-DP1.app/Contents/Developer/usr/bin/git
> xcrun git --version
git version 1.7.9.6 (Apple Git-31)
>

Using `xcrun`


> xcrun -find git
/Applications/Xcode45-DP1.app/Contents/Developer/usr/bin/git
> xcrun git --version
git version 1.7.9.6 (Apple Git-31)
> alias git="xcrun git"

Using `xcrun`


> xcrun -find git
/Applications/Xcode45-DP1.app/Contents/Developer/
usr/bin/git
> xcrun git --version
git version 1.7.9.6 (Apple Git-31)
> alias git="xcrun git"
>

Using `xcrun`

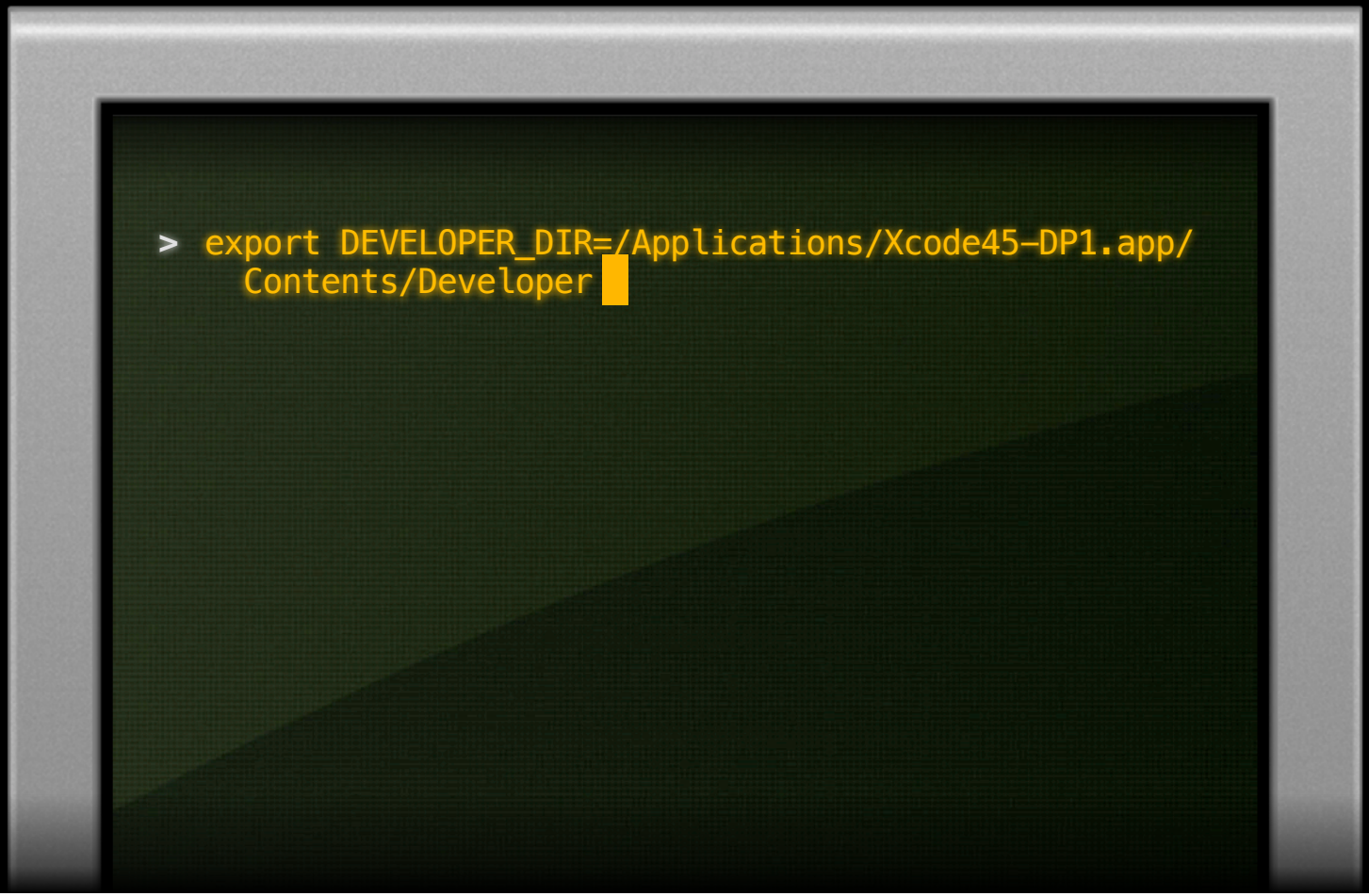
```
> xcrun -find git
/Applications/Xcode45-DP1.app/Contents/Developer/
usr/bin/git

> xcrun git --version
git version 1.7.9.6 (Apple Git-31)

> alias git="xcrun git"

> git --version
```

Using `xcrun`

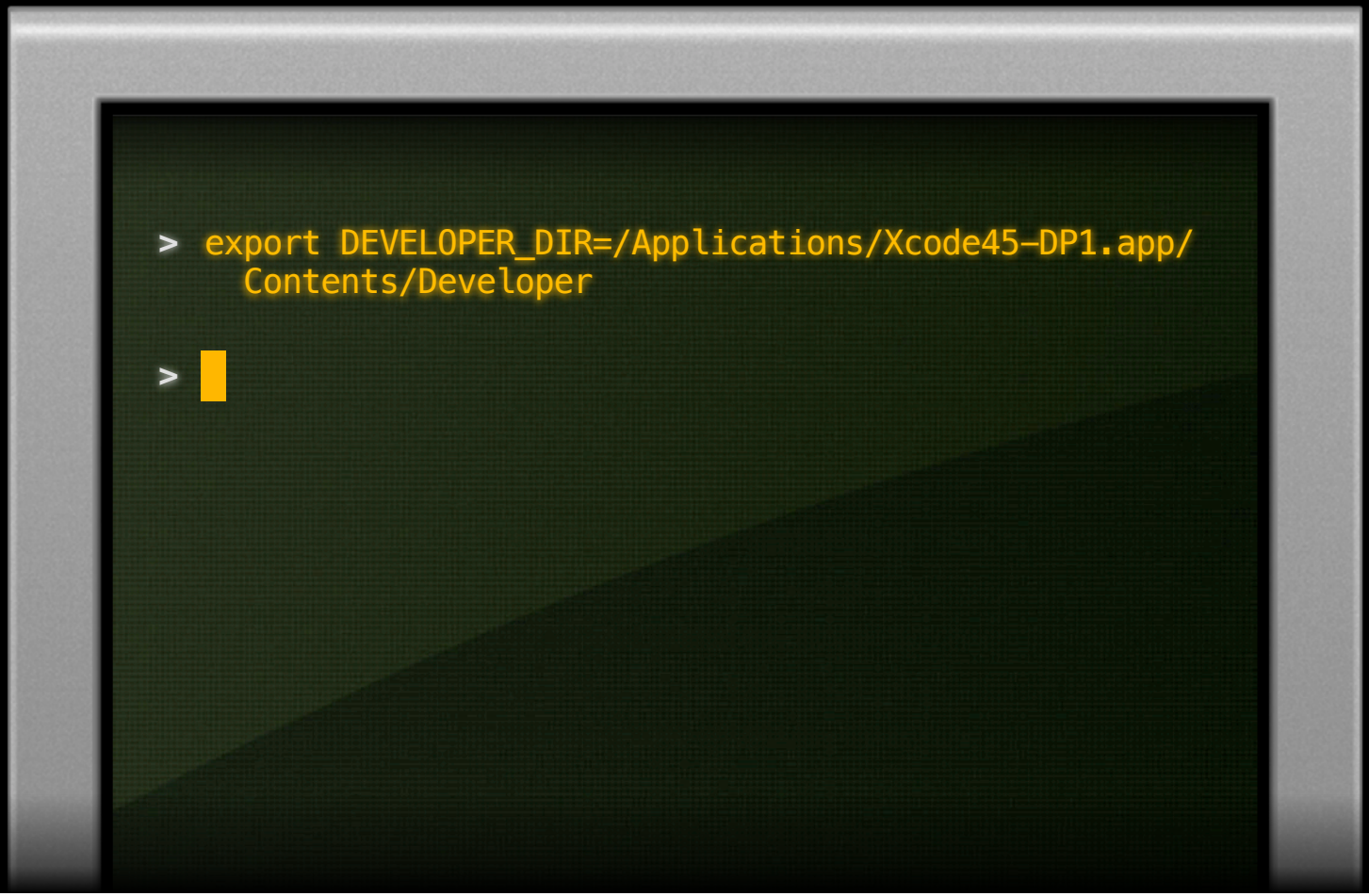
```
> xcrun -find git
/Applications/Xcode45-DP1.app/Contents/Developer/
usr/bin/git

> xcrun git --version
git version 1.7.9.6 (Apple Git-31)

> alias git="xcrun git"

> git --version
git version 1.7.9.6 (Apple Git-31)

>
```

Directly Invoking a Specific `xcodebuild`

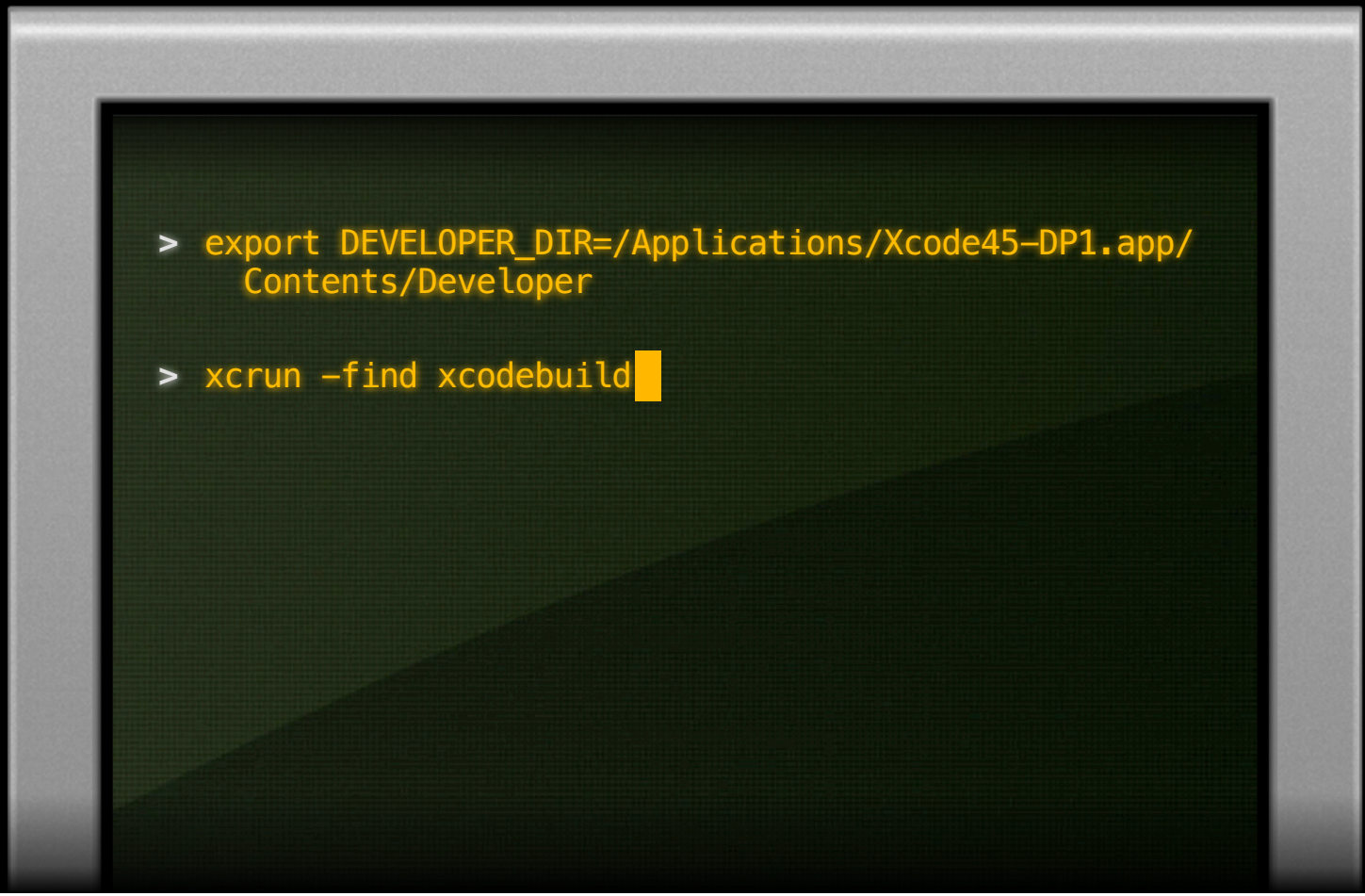
Directly Invoking a Specific `xcodebuild`



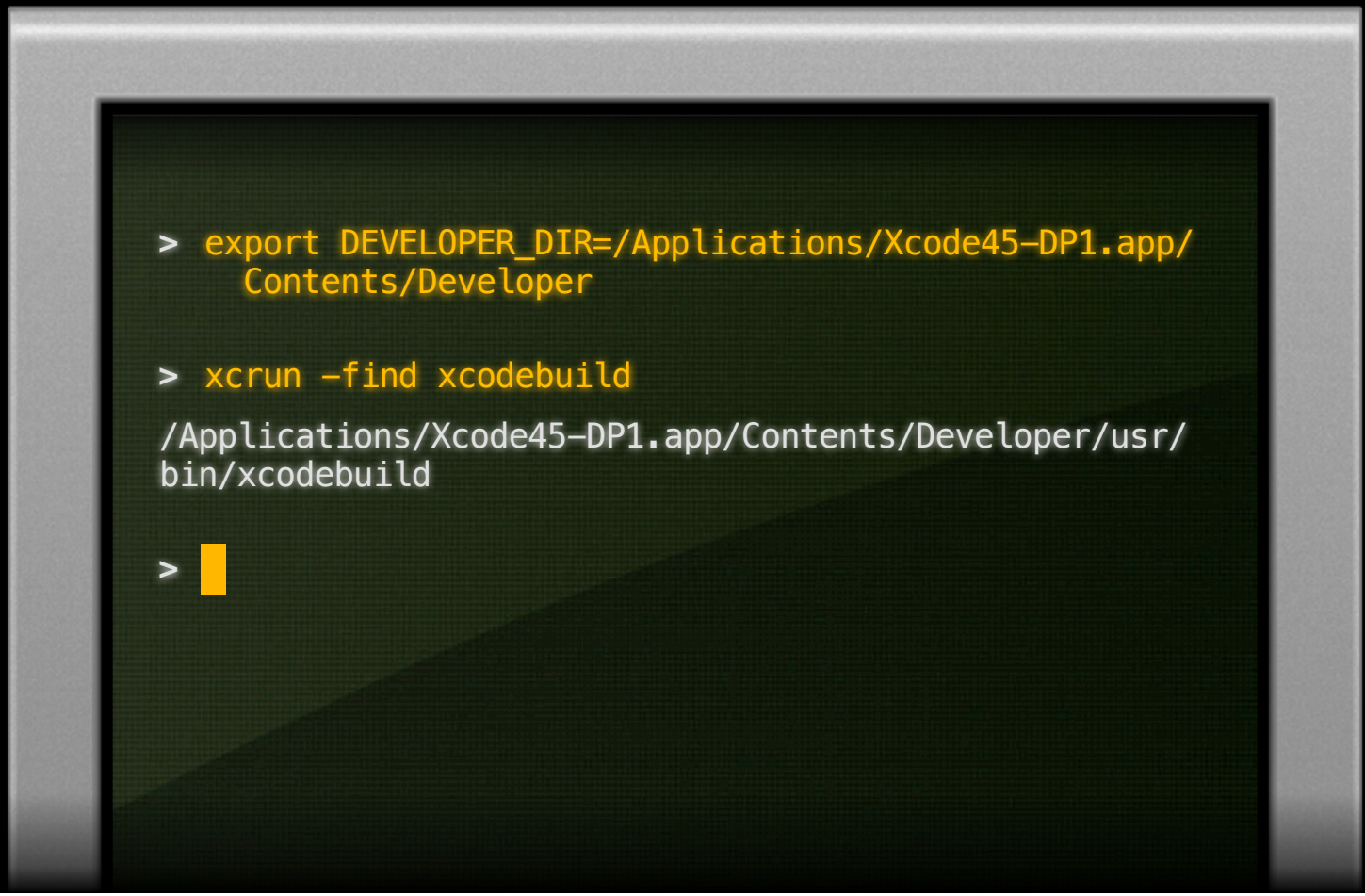
Directly Invoking a Specific `xcodebuild`



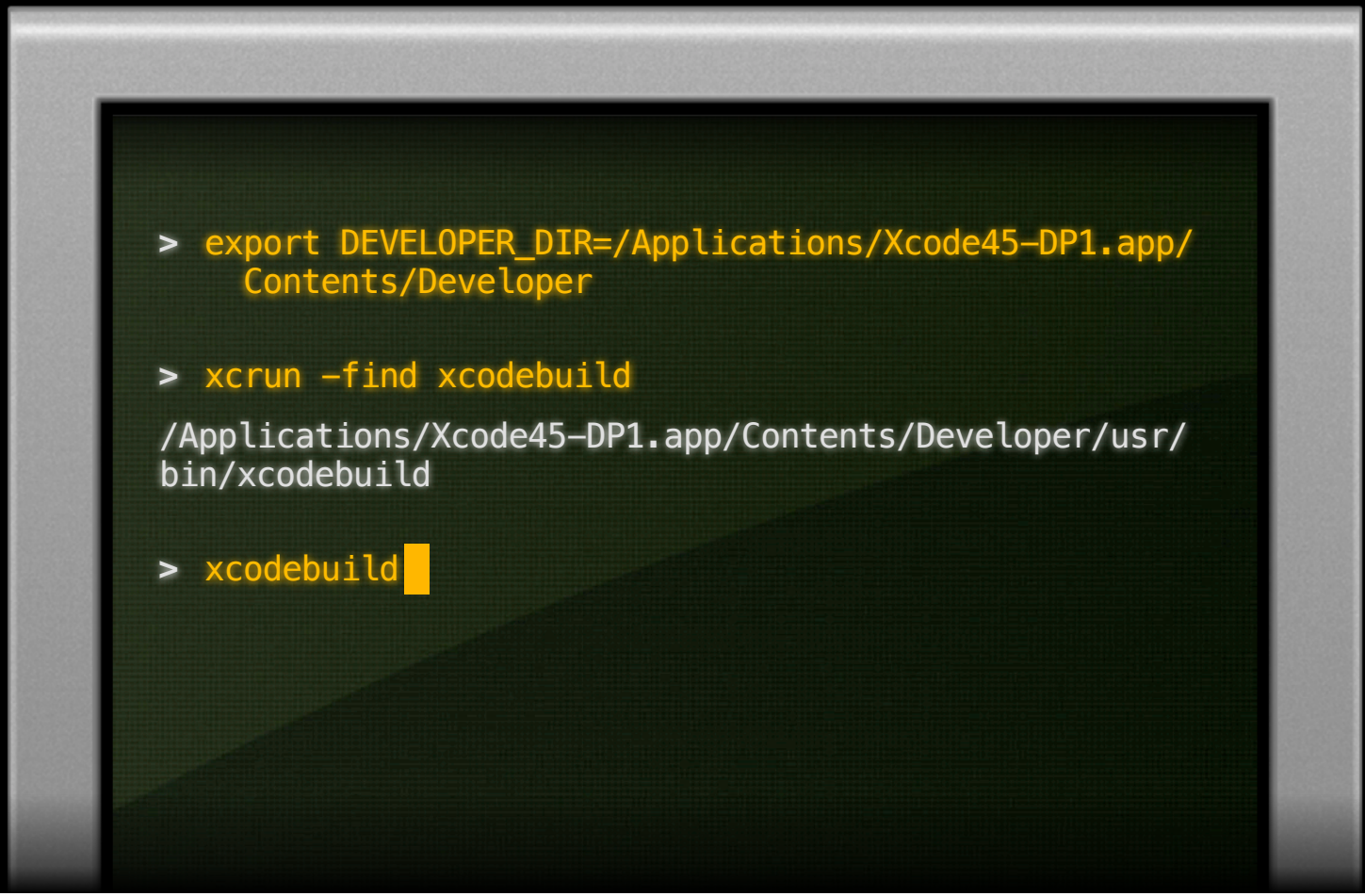
Directly Invoking a Specific `xcodebuild`



Directly Invoking a Specific `xcodebuild`



Directly Invoking a Specific `xcodebuild`



Directly Invoking a Specific xcodebuild

```
> export DEVELOPER_DIR=/Applications/Xcode45-DP1.app/
    Contents/Developer

> xcrun -find xcodebuild

/Applications/Xcode45-DP1.app/Contents/Developer/usr/
bin/xcodebuild

> xcodebuild

=== BUILD NATIVE TARGET Baffle OSX OF PROJECT Baffle
OSX WITH CONFIGURATION Debug ===

ProcessPCH build/Intermediates/PrecompiledHeaders/Baff
leCocoa_Prefix-dkhujustkymhfqbdihrzilncxtxa/BaffleCoco
a_Prefix.pch.pth OtherSources/BaffleCocoa_Prefix.pch n
ormal x86_64 objective-c com.apple.compilers.llvm.clan
```

Automating `xcodebuild`

Using Xcode with Continuous Integration

Daniel Dunbar
System Tools Engineer

What Is Continuous Integration?

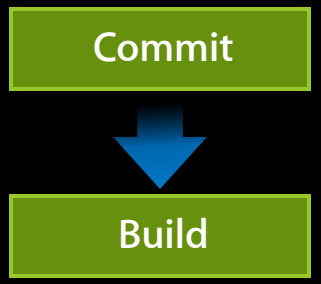
- Continuous Integration (CI) refers to a continuous process for integrating and applying quality checks to software development

What Is Continuous Integration?

- Continuous Integration (CI) refers to a continuous process for integrating and applying quality checks to software development

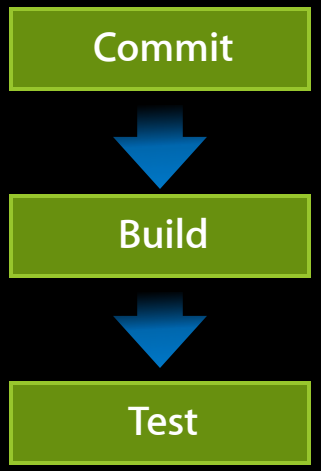
What Is Continuous Integration?

- Continuous Integration (CI) refers to a continuous process for integrating and applying quality checks to software development



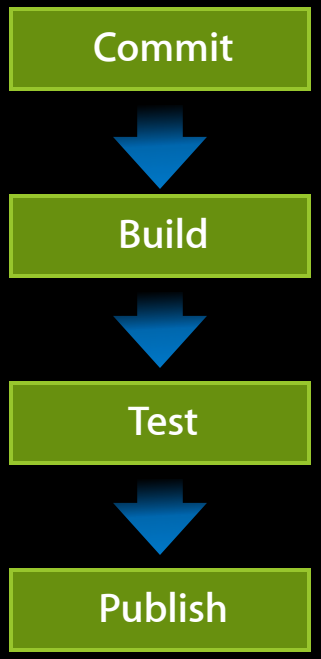
What Is Continuous Integration?

- Continuous Integration (CI) refers to a continuous process for integrating and applying quality checks to software development



What Is Continuous Integration?

- Continuous Integration (CI) refers to a continuous process for integrating and applying quality checks to software development



Continuous Integration Example

Continuous Integration Example

- Common problem

Continuous Integration Example

- Common problem
  - Develop with Xcode GM and Xcode WWDC Preview tools

Continuous Integration Example

- Common problem
  - Develop with Xcode GM and Xcode WWDC Preview tools
  - Code shared between two products (e.g. OS X and iOS)

Continuous Integration Example

- Common problem
  - Develop with Xcode GM and Xcode WWDC Preview tools
  - Code shared between two products (e.g. OS X and iOS)
  - Disruptive to switch between tools constantly

Continuous Integration Example

- Common problem
  - Develop with Xcode GM and Xcode WWDC Preview tools
  - Code shared between two products (e.g. OS X and iOS)
  - Disruptive to switch between tools constantly
- Solution

Continuous Integration Example

- Common problem
  - Develop with Xcode GM and Xcode WWDC Preview tools
  - Code shared between two products (e.g. OS X and iOS)
  - Disruptive to switch between tools constantly
- Solution
  - Use `xcodebuild` and Continuous Integration

Automating `xcodebuild`

Automating `xcodebuild`

- Set `DEVELOPER_DIR` before running `xcodebuild`

Automating `xcodebuild`

- Set `DEVELOPER_DIR` before running `xcodebuild`
  - This ensures the Xcode tools you want are used, regardless of the tools currently chosen by `xcode-select`

Automating `xcodebuild`

- Set `DEVELOPER_DIR` before running `xcodebuild`
  - This ensures the Xcode tools you want are used, regardless of the tools currently chosen by `xcode-select`
- Use appropriate options to set the workspace, scheme, SDK

Automating `xcodebuild`

- Set `DEVELOPER_DIR` before running `xcodebuild`
  - This ensures the Xcode tools you want are used, regardless of the tools currently chosen by `xcode-select`
- Use appropriate options to set the workspace, scheme, SDK
- Use build settings to control where output files go

Automating `xcodebuild`

- Set `DEVELOPER_DIR` before running `xcodebuild`
  - This ensures the Xcode tools you want are used, regardless of the tools currently chosen by `xcode-select`
- Use appropriate options to set the workspace, scheme, SDK
- Use build settings to control where output files go

`DSTROOT, OBJROOT, SYMROOT, SHARED_PRECOMPS_DIR`

Automating `xcodebuild`

Automating `xcodebuild`

- Make sure builds are run as a user who has accepted the license agreement; alternately, can accept for all users

```
sudo xcodebuild -license
```

Automating `xcodebuild`

- Make sure builds are run as a user who has accepted the license agreement; alternately, can accept for all users

```
sudo xcodebuild –license
```

- Ensure access to appropriate development certificates

Demo

Automating project builds

Working Outside Xcode

Working Outside Xcode

- Many projects may not supply native Xcode projects

Working Outside Xcode

- Many projects may not supply native Xcode projects
  - Legacy code base

Working Outside Xcode

- Many projects may not supply native Xcode projects
  - Legacy code base
  - Traditional UNIX development

Working Outside Xcode

- Many projects may not supply native Xcode projects
  - Legacy code base
  - Traditional UNIX development
  - Cross-platform open source projects

Working Outside Xcode

- Many projects may not supply native Xcode projects
  - Legacy code base
  - Traditional UNIX development
  - Cross-platform open source projects
- Tools for working outside Xcode

Working Outside Xcode

- Many projects may not supply native Xcode projects
  - Legacy code base
  - Traditional UNIX development
  - Cross-platform open source projects
- Tools for working outside Xcode
  - Understanding the command line tools package

The Command Line Tools Package

The Command Line Tools Package

- What is the command line tools package?

The Command Line Tools Package

- What is the command line tools package?
- How to install the command line tools package?

The Command Line Tools Package

- What is the command line tools package?
- How to install the command line tools package?
- When should it be installed?

What Is the Package?

What Is the Package?

- New in Xcode 4.3

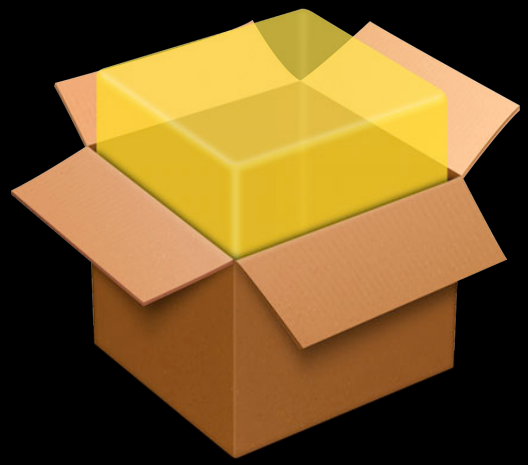
What Is the Package?

- New in Xcode 4.3
- Separate package for doing extensive command line development

What Is the Package?

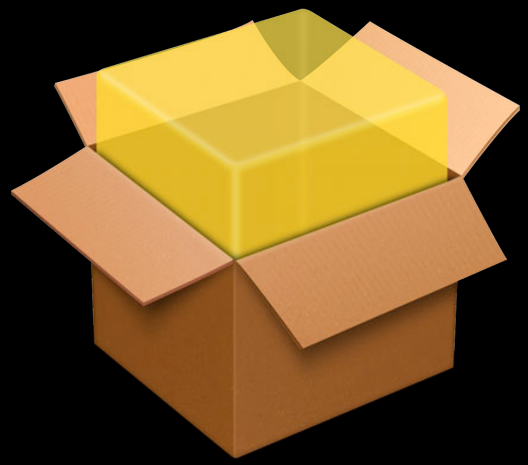
- New in Xcode 4.3
- Separate package for doing extensive command line development
- Supported mechanism for UNIX-style development

What Does It Contain?



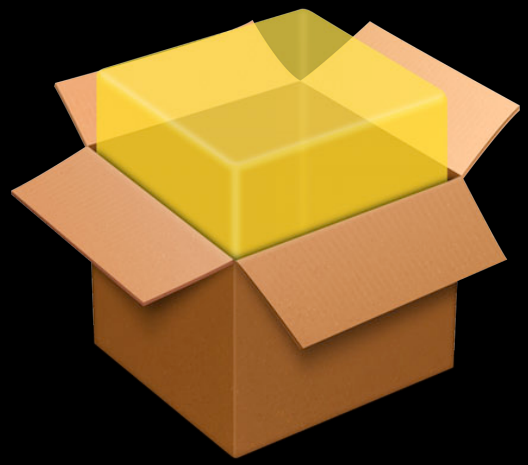
What Does It Contain?

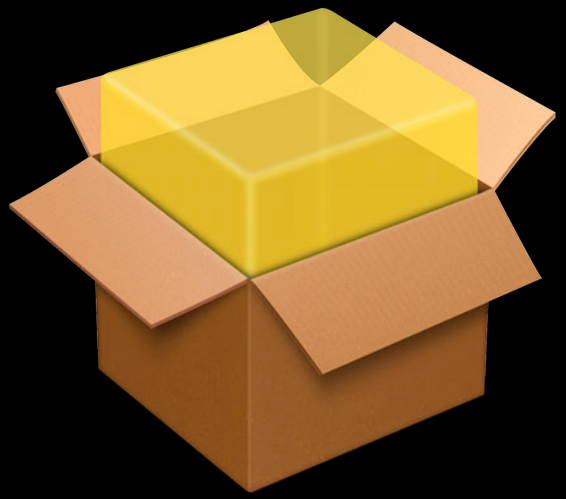
- Contains everything for command line development

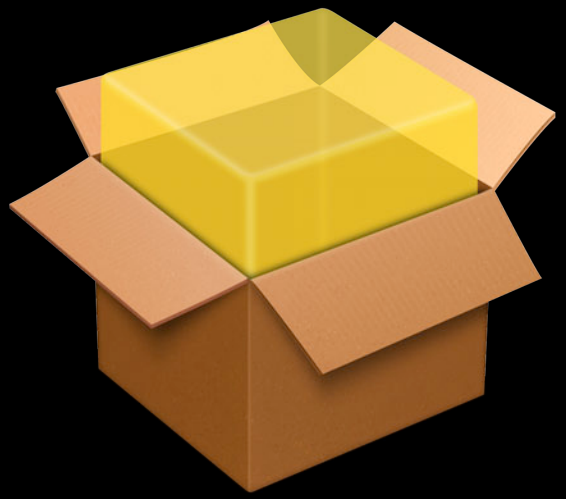


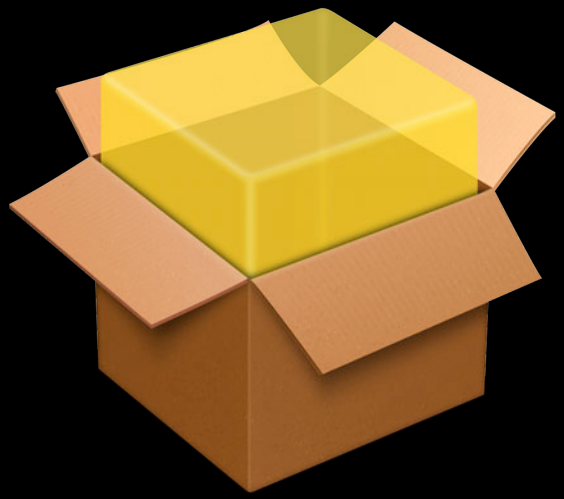
What Does It Contain?

- Contains everything for command line development
- Bundled at the standard UNIX location (/usr)





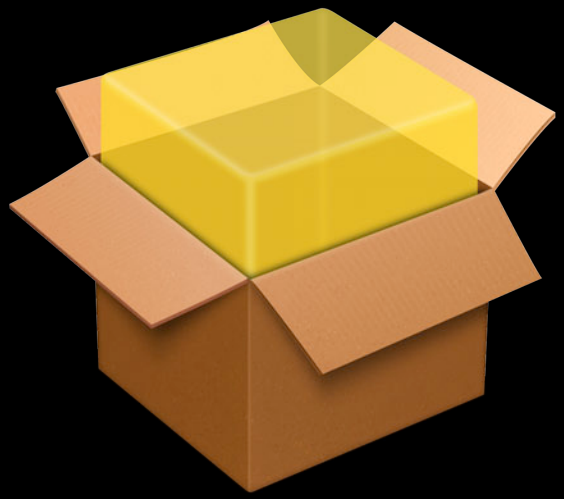




Tools

OSX / SDK

Documentation
(man Pages)



Tools

OSX / SDK

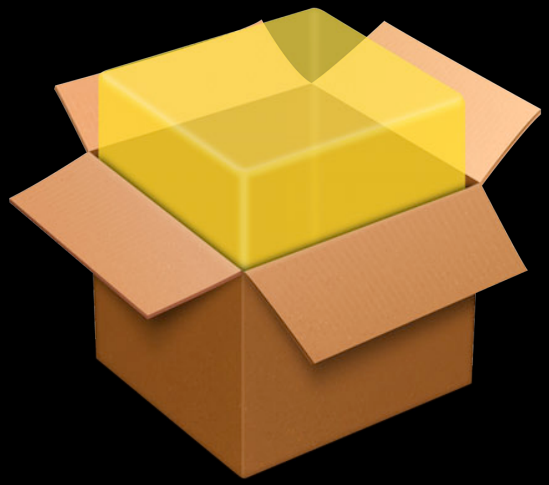
Documentation
(man Pages)

Development Tools

Debugging Tools

Source Control

Obtaining the Command Line Tools



Download from
Developer Resources

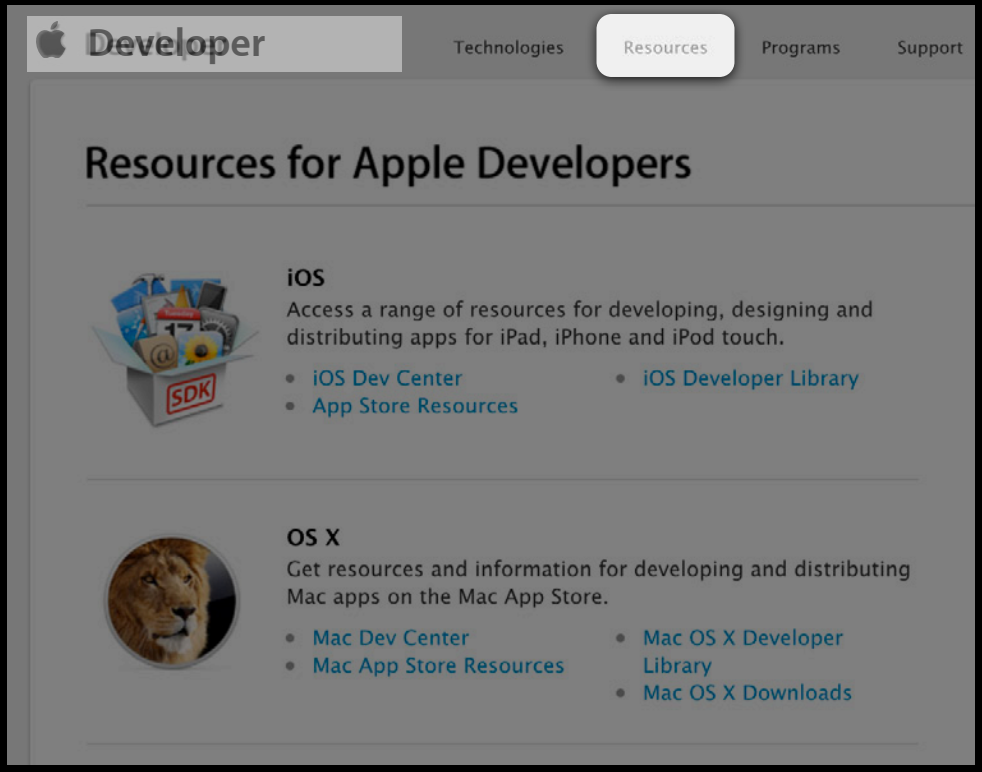
Install from Xcode

Obtaining the Command Line Tools

As a standalone download

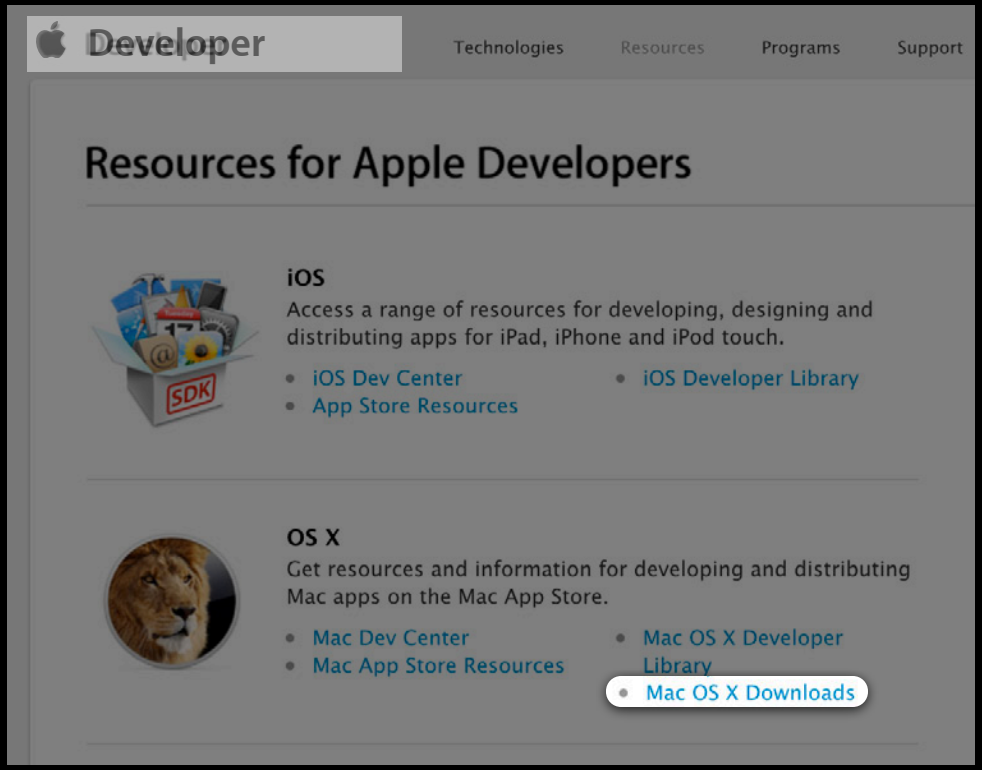
Obtaining the Command Line Tools

As a standalone download



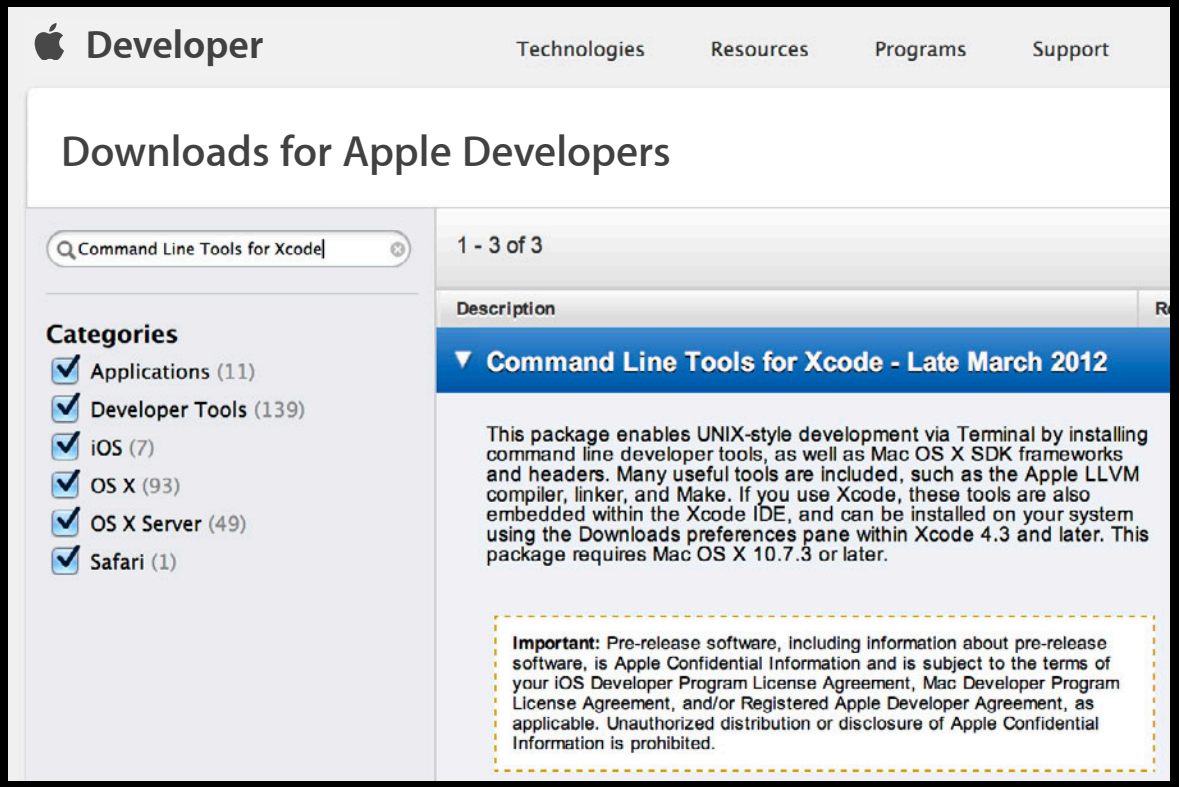
Obtaining the Command Line Tools

As a standalone download



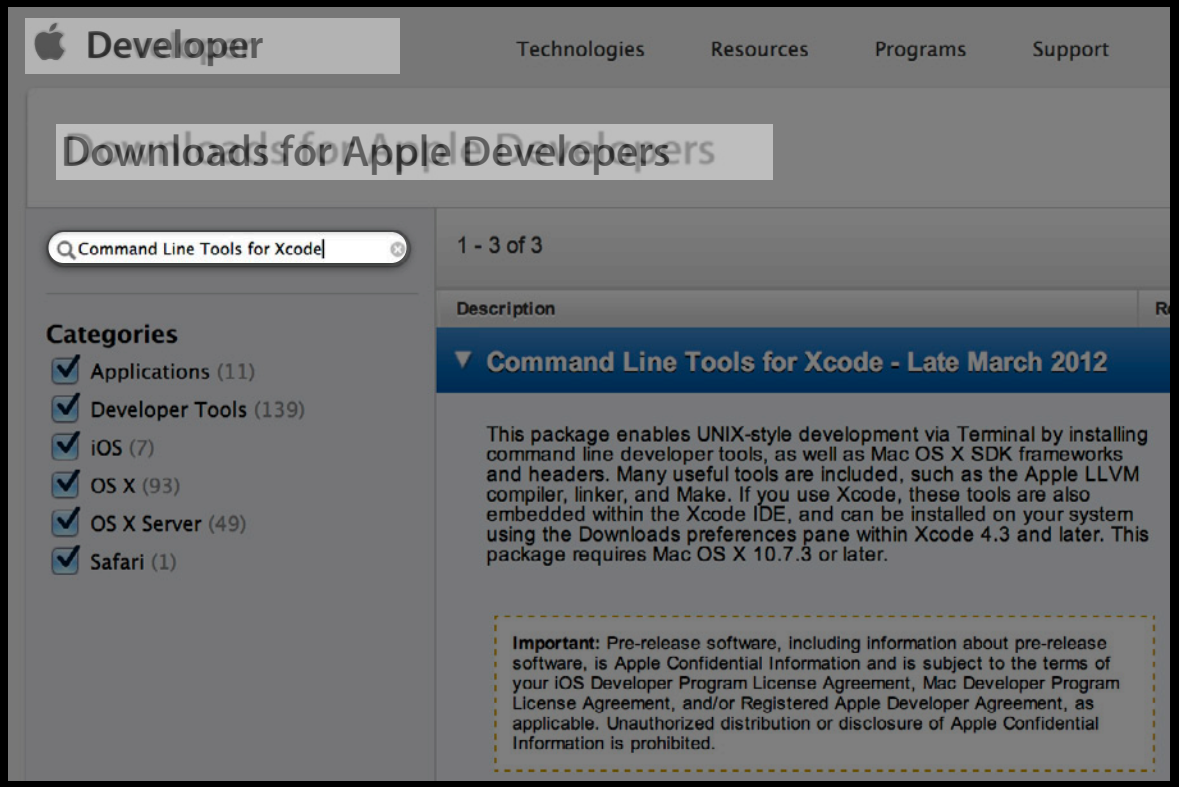
Obtaining the Command Line Tools

As a standalone download



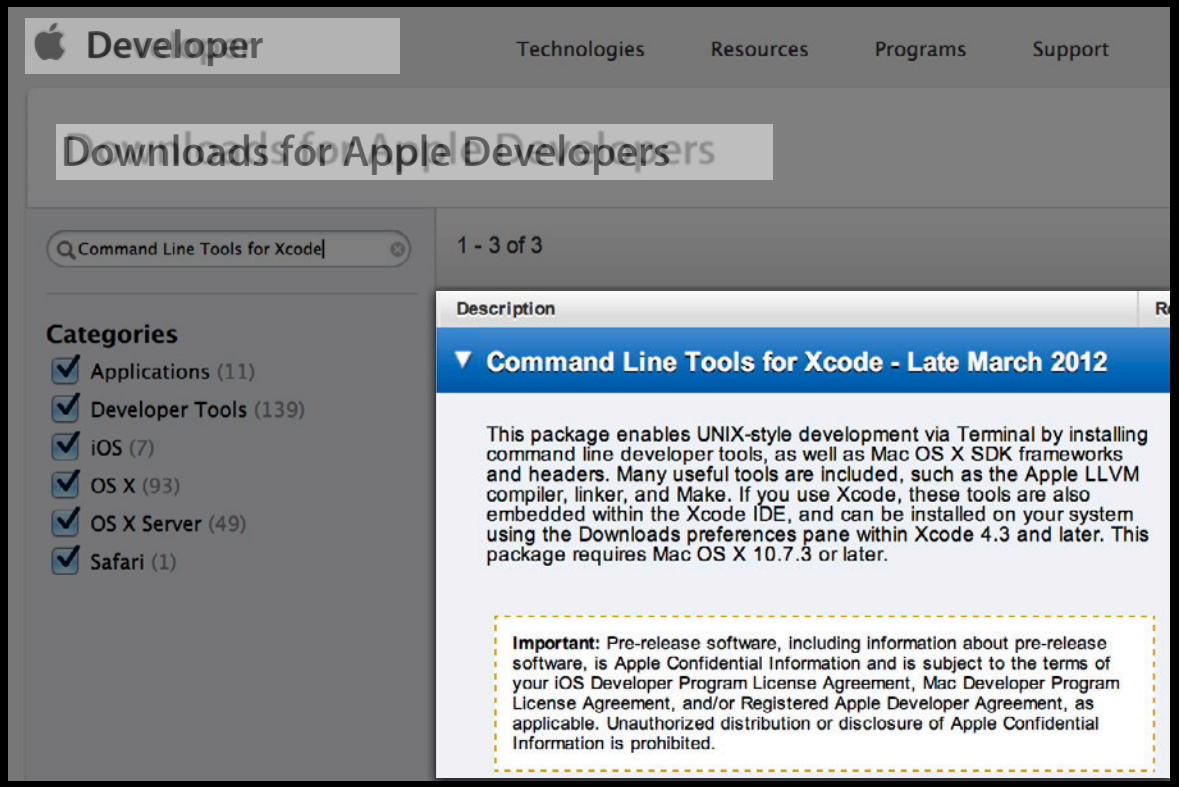
Obtaining the Command Line Tools

As a standalone download



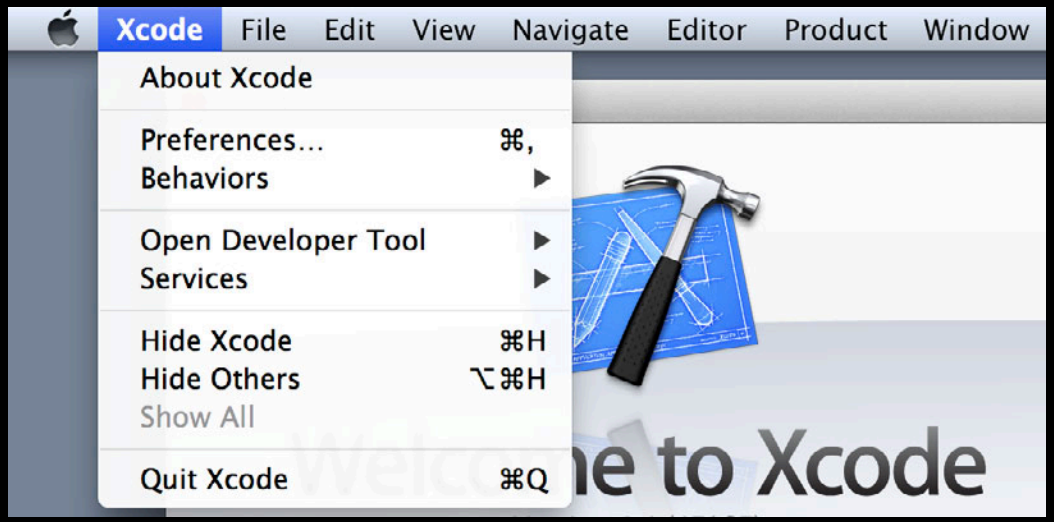
Obtaining the Command Line Tools

As a standalone download



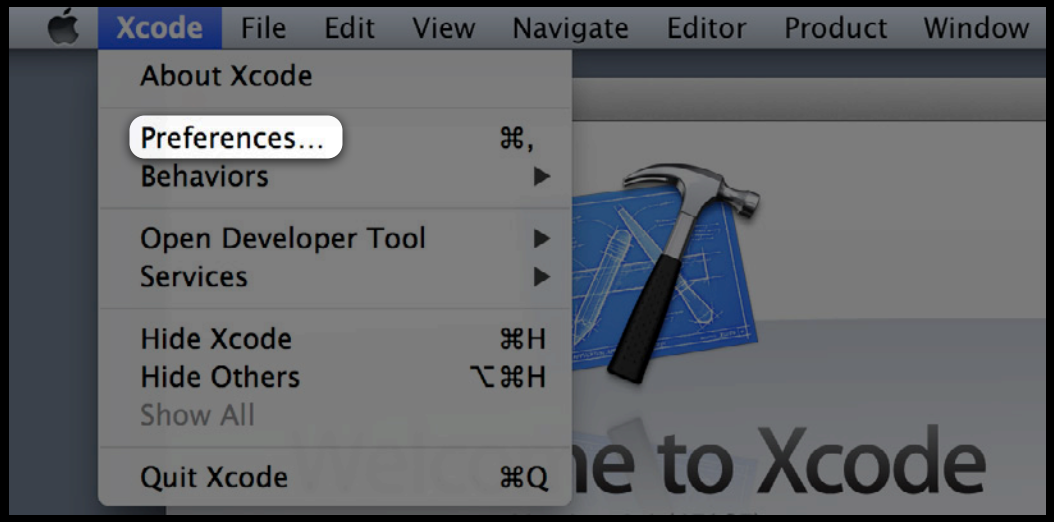
Obtaining the Command Line Tools

As part of Xcode



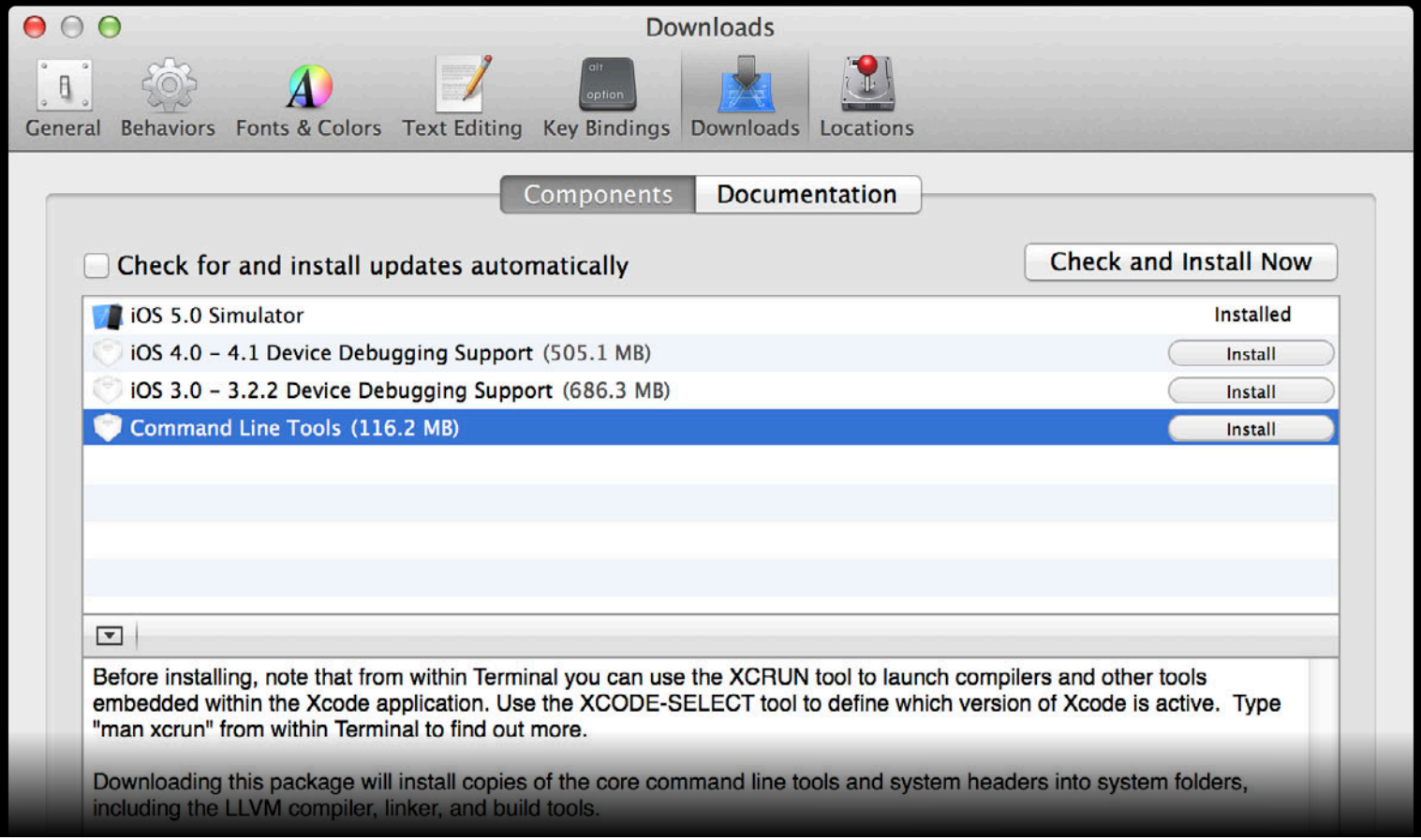
Obtaining the Command Line Tools

As part of Xcode



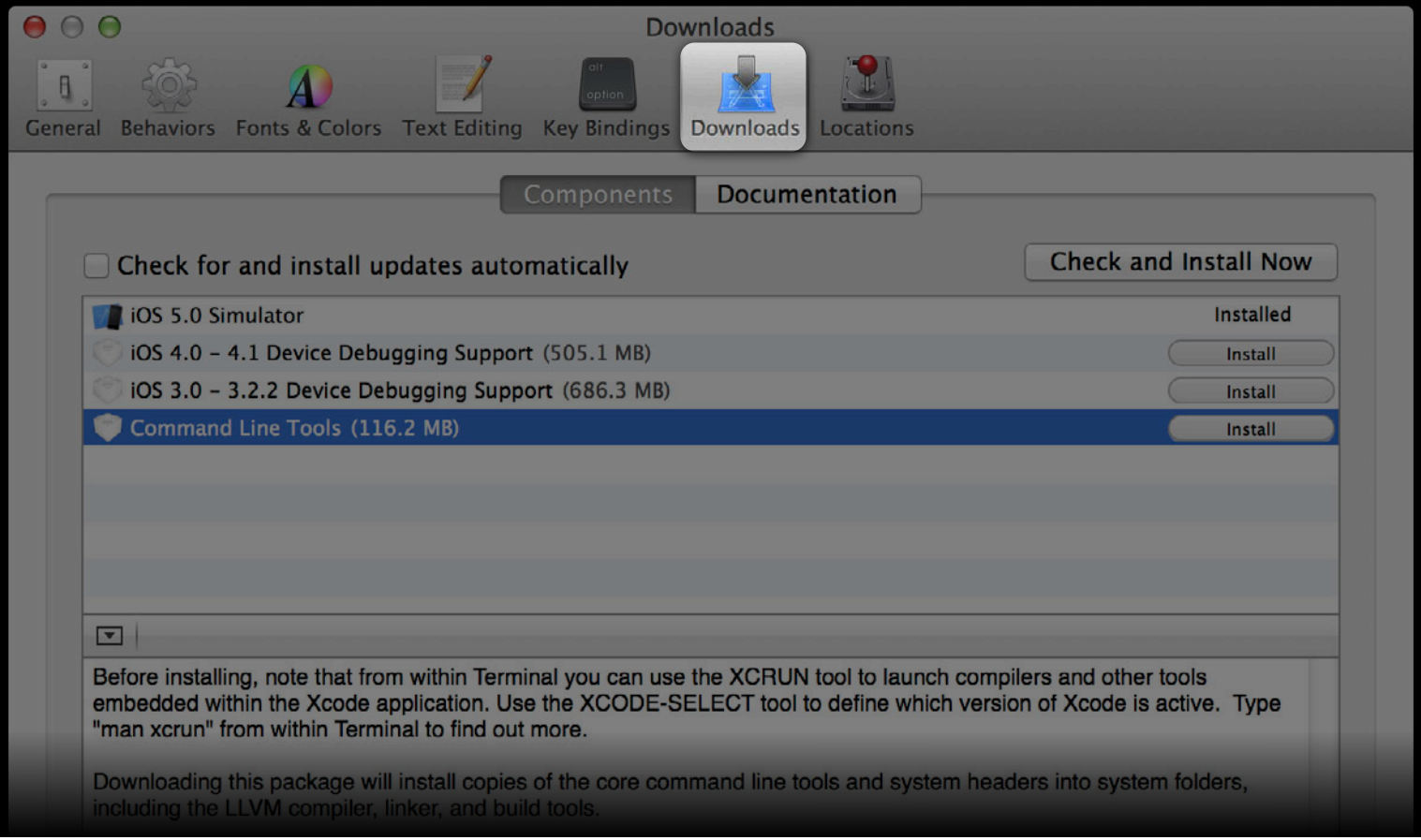
Obtaining the Command Line Tools

As part of Xcode



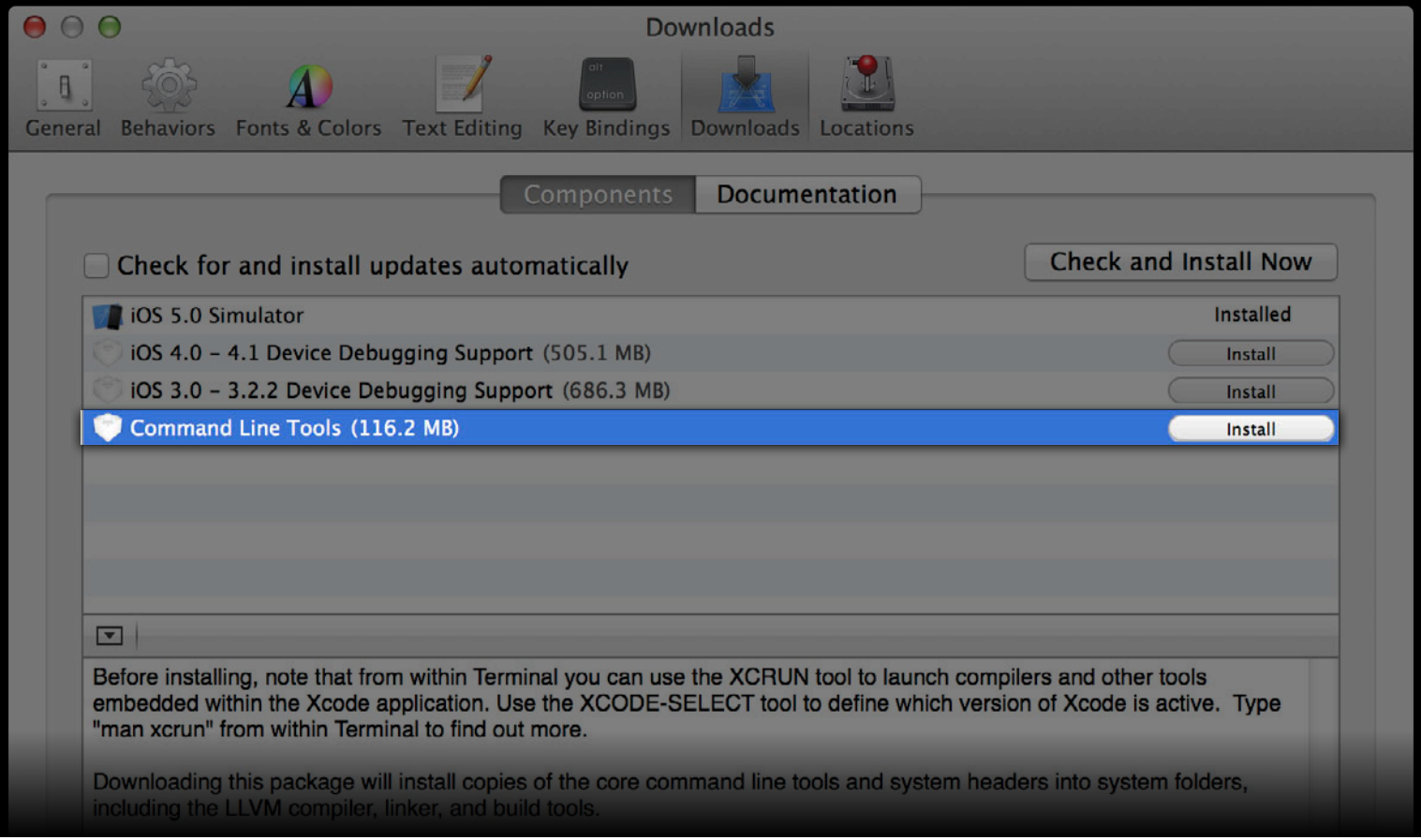
Obtaining the Command Line Tools

As part of Xcode



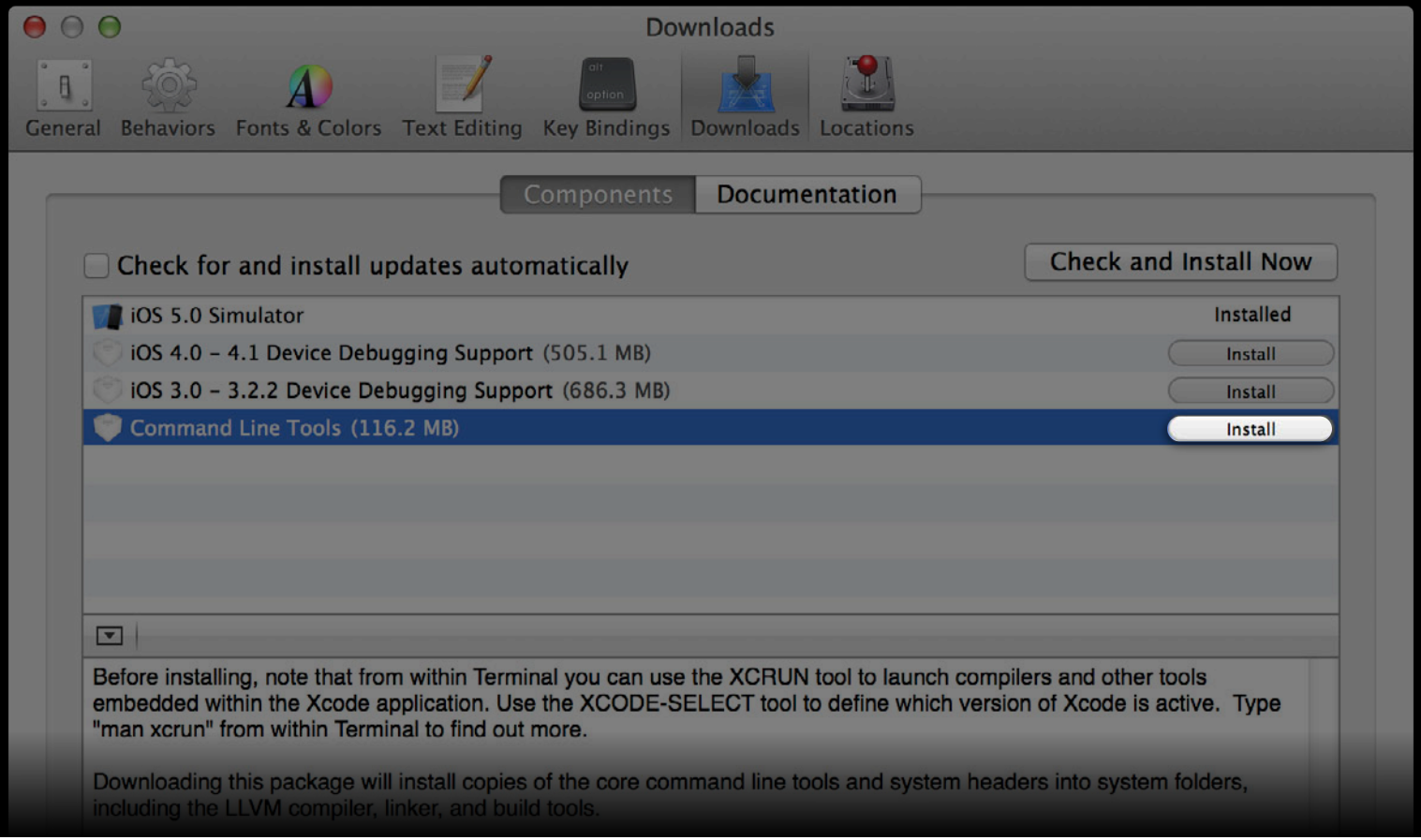
Obtaining the Command Line Tools

As part of Xcode



Obtaining the Command Line Tools

As part of Xcode



When to Install the Package

- Open source development

When to Install the Package

- Open source development
  - Recommended practice for working with Fink, MacPorts, Homebrew

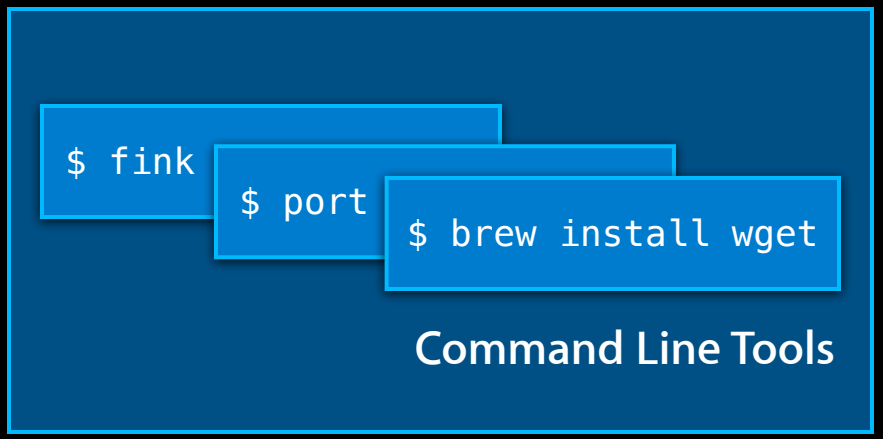
When to Install the Package

- Open source development
  - Recommended practice for working with Fink, MacPorts, Homebrew

```
$ fink
$ port
$ brew install wget
```

When to Install the Package

- Open source development
  - Recommended practice for working with Fink, MacPorts, Homebrew



When to Install the Package

- Extensive command line usage

When to Install the Package

- Extensive command line usage
  - Traditional UNIX-style workflow

When to Install the Package

- Extensive command line usage
  - Traditional UNIX-style workflow
  - Legacy projects

When to Install the Package

- Extensive command line usage
  - Traditional UNIX-style workflow
  - Legacy projects
  - Non-Xcode IDE based development (e.g., Makefiles)

When to Install the Package

- Limited development, small download size

When to Install the Package

- Limited development, small download size
  - Command line tools package is self-contained

When to Install the Package

- Limited development, small download size
  - Command line tools package is self-contained
  - Does not contain Xcode IDE or other packages

When Not to Install the Package

When Not to Install the Package

- Do not install just to get `svn` or `git`

When Not to Install the Package

- Do not install just to get `svn` or `git`
  - Instead, use `xcrun` and shell aliases

When Not to Install the Package

- Do not install just to get `svn` or `git`
  - Instead, use `xcrun` and shell aliases
- Do not install unless you need them

When Not to Install the Package

- Do not install just to get `svn` or `git`
  - Instead, use `xcrun` and shell aliases
- Do not install unless you need them
  - Only one instance of command line tools

When Not to Install the Package

- Do not install just to get `svn` or `git`
  - Instead, use `xcrun` and shell aliases
- Do not install unless you need them
  - Only one instance of command line tools
  - Install is distinct from Xcode

When Not to Install the Package

- Do not install just to get `svn` or `git`
  - Instead, use `xcrun` and shell aliases
- Do not install unless you need them
  - Only one instance of command line tools
  - Install is distinct from Xcode
  - Requires manual effort to keep up-to-date with new releases

Review

Primary Xcode Command Line Tools

Tool	Summary
`xcode-select`	Choose the default Xcode.app to use
`xcodebuild`	Build, archive, and query Xcode projects and workspaces
`xcrun`	Find and execute command line tools that are inside Xcode
`man`	Access to command line documentation

Common `xcode-select` Options

Option	Usage
`-print-path`	Show the currently selected Xcode Developer directory
`-switch <path>`	Switch to the given Xcode (or Developer directory)

Common `xcodebuild` Options

Option	Usage
`-workspace <name>` `-project <name>`	Select a project or workspace to operate on
`-scheme <name>`	Select a scheme to use
`-sdk <name>`	Select an SDK to use
`-showsdks`	List available SDKs
`-list`	List schemes or targets for active project or workspace
`-showBuildSettings`	List active build settings and their values
`NAME=VALUE`	Override a build setting with a particular value

Common xcrun Options

Option	Usage
`-find <name>`	Find a tool in the active Xcode
`-sdk <name>`	Select an SDK to perform lookups in
`<tool> ... args ...`	Find and execute the tool with the provided arguments

More Information

Michael Jurewitz
Developer Tools Evangelist
jury@apple.com

Documentation
Xcode Build Setting Reference
http://developer.apple.com

man Pages
http://developer.apple.com
(or `man <toolname>` on the command line)

Apple Developer Forums
http://devforums.apple.com

Related Sessions

Working with Schemes and Projects in Xcode	Marina Wednesday 4:30PM
Building, Archiving, and Submitting Your App	Pacific Heights Thursday 4:30PM

Labs

Xcode Lab	Developer Tools Lab B Daily 9:00AM
Xcode Lab	Developer Tools Lab B Daily 2:00PM

Summary

Summary

- Xcode provides a number of tools for working on the command

Summary

- Xcode provides a number of tools for working on the command
- Use `xcodebuild` to script and automate builds

Summary

- Xcode provides a number of tools for working on the command
- Use `xcodebuild` to script and automate builds
  - Use build settings and schemes to customize to your project

Summary

- Xcode provides a number of tools for working on the command
- Use `xcodebuild` to script and automate builds
  - Use build settings and schemes to customize to your project
- Consider using continuous integration to improve development

Summary

- Xcode provides a number of tools for working on the command
- Use `xcodebuild` to script and automate builds
  - Use build settings and schemes to customize to your project
- Consider using continuous integration to improve development
- Use the Command Line Tools package when working outside Xcode


WWDC2012