Agenda
Agenda

Testing Concepts
Agenda

Testing Concepts

Xcode
  • Crash Log Gathering
Agenda

Testing Concepts

Xcode
• Crash Log Gathering

Xcode Server
• Advanced Triggers
• Issue Tracking and Blame
• Configurable Integration User
Agenda

Testing Concepts

Xcode
  • Crash Log Gathering

Xcode Server
  • Advanced Triggers
  • Issue Tracking and Blame
  • Configurable Integration User

xcodebuild
Overview
Overview
XCTest
Testing framework
XCTest

Testing framework

Sources

TestClassA.m

TestClassB.swift

TestClassC.swift
XCTest
Testing framework

Sources
- TestClassA.m
- TestClassB.swift
- TestClassC.swift

Build Products
- UI.xctest
- Unit.xctest

Compile
Xcode

Test harness
Xcode

Test harness

Author tests
Xcode

Test harness

Author tests

Execute tests
Xcode

Test harness

Author tests
Execute tests
Review reports
Continuous Integration

Xcode Server
Xcode Server
Continuous Integration

Schedules bot runs
Xcode Server
Continuous Integration

Schedules bot runs
Generates reports
Xcode Server
Continuous Integration

Schedules bot runs
Generates reports
Sends notifications
xcodebuild

Command line tool
xcodebuild
Command line tool

Used by Xcode Server

```
Test Suite 'Selected tests' started at 2016-05-13 16:52:30.048
Test Suite 'ExampleUITests' started at 2016-05-13 16:52:30.049
Test Case '-[ExampleUITests.ExampleUITests testExample]' started.
t = 0.00s Start Test at 2016-05-13 16:52:30.050
  t = 0.00s Set Up
  t = 0.00s Launch com.toolsQA.Example
  t = 1.32s Waiting for accessibility to load
  t = 1.73s Wait for app to idle
  t = 3.05s Tear Down
Test Case '-[ExampleUITests.ExampleUITests testExample]' passed (3.055 seconds).
Test Suite 'ExampleUITests' passed at 2016-05-13 16:52:33.106.
  Executed 1 test, with 0 failures (0 unexpected) in 3.055 (3.057) seconds
Test Suite 'Selected tests' passed at 2016-05-13 16:52:33.106.
  Executed 1 test, with 0 failures (0 unexpected) in 3.055 (3.058) seconds
** TEST SUCCEEDED **
```

Zoltans-MacBook:Example zoltan$
xcodebuild
Command line tool

Used by Xcode Server
Build and execute tests
xcodebuild
Command line tool

Used by Xcode Server
Build and execute tests
For custom CI systems

```
Test Suite 'Selected tests' started at 2016-05-13 16:52:30.048
Test Suite 'ExampleUITests' started at 2016-05-13 16:52:30.049
Test Case '[-ExampleUITests.ExampleUITests testExample]' started.
t = 0.00s Start Test at 2016-05-13 16:52:30.050
  t = 0.00s Set Up
  t = 0.00s Launch com.toolsQA.Example
  t = 1.32s Waiting for accessibility to load
  t = 1.73s Wait for app to idle
  t = 3.05s Tear Down
Test Case '[-ExampleUITests.ExampleUITests testExample]' passed (3.055 seconds).
Test Suite 'ExampleUITests' passed at 2016-05-13 16:52:33.106.
  Executed 1 test, with 0 failures (0 unexpected) in 3.055 (3.057) seconds
Test Suite 'Selected tests' passed at 2016-05-13 16:52:33.106.
  Executed 1 test, with 0 failures (0 unexpected) in 3.055 (3.058) seconds
** TEST SUCCEEDED **
```

Zoltans-MacBook:Example zoltan$
## Previous Years’ Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>WWDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing with Xcode 6</td>
<td>2014</td>
</tr>
<tr>
<td>Continuous Integration with Xcode 6</td>
<td>2014</td>
</tr>
<tr>
<td>UI Testing in Xcode</td>
<td>2015</td>
</tr>
<tr>
<td>CI and Code Coverage in Xcode</td>
<td>2015</td>
</tr>
</tbody>
</table>
Testing Concepts
How Testing Works

Behind the scenes

MyTests.xctest

MyApp.app
Testing Timeline

Fundamental concepts

Compilation  Hosting  Observation
<table>
<thead>
<tr>
<th>Unit Tests</th>
<th>UI Tests</th>
</tr>
</thead>
</table>

How testing starts
Test Hosting
How testing starts

- Unit Tests
- UI Tests

MyApp.app
Test Hosting
How testing starts

Unit Tests

MyApp.app

UI Tests
Test Hosting
How testing starts

<table>
<thead>
<tr>
<th>Unit Tests</th>
<th>UI Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>MyApp.app</td>
<td>Host Application</td>
</tr>
</tbody>
</table>
Test Hosting
How testing starts

Unit Tests

MyApp.app

Host Application

UI Tests

UI Test Runner

MyApp.app
Test Hosting
How testing starts

<table>
<thead>
<tr>
<th>Unit Tests</th>
<th>UI Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>MyApp.app</td>
<td>UI Test Runner</td>
</tr>
</tbody>
</table>

Host Application
Test Hosting
How testing starts

Unit Tests

Host Application
MyApp.app

UI Tests

UI Test Runner
MyApp.app

Target Application
## Test Hosting Comparison

<table>
<thead>
<tr>
<th>Unit Tests</th>
<th>UI Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct access to host app data and APIs</td>
<td>Uses Accessibility to access target app</td>
</tr>
</tbody>
</table>
## Test Hosting

### Comparison

<table>
<thead>
<tr>
<th>Unit Tests</th>
<th>UI Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct access to host app data and APIs</td>
<td>Uses Accessibility to access target app</td>
</tr>
<tr>
<td>All tests run in single app launch</td>
<td>Tests launch app for every test case</td>
</tr>
</tbody>
</table>
Testing Timeline
Fundamental concepts

Compilation  Hosting
Testing Timeline

Fundamental concepts

Compilation  Hosting  Observation
Test Observation

Test case structure
Test Observation
Test case structure
Test Observation

Test case structure
Test Observation

Test case structure

Test Suite A

Test Suite B
Test Observation
Test case structure

Test Suite A

Test Bundle

Test Suite B
Test Observation

Test case structure

Setup or tear down work
Test Observation

Test case structure

Setup or tear down work

Custom logging

Test Bundle

Test Suite A

Test Suite B
optional public func testBundleWillStart(_ testBundle: AnyObject!)
optional public func testSuiteWillStart(_ testSuite: AnyObject!)
optional public func testCaseWillStart(_ testCase: AnyObject!)
optional public func testCaseDidFinish(_ testCase: AnyObject!)
optional public func testCase(_ testCase: AnyObject!, didFailWithDescription description: AnyObject!, inFile filePath: AnyObject!, atLine lineNumber: AnyObject!)

Test Observation

XCTestObservation Protocol

CASE DID FAIL

Test Bundle

Test Suite A

Test Suite B
optional public func testSuiteDidFinish(_ testSuite: AnyObject!)
optional public func testBundleDidFinish(_ testBundle: AnyObject!)

Test Observation
XCTestObservation Protocol

optional public func testBundleDidFinish(_ testBundle: AnyObject!)
// Test Observation
// Example LoggingObserver

class LoggingObserver: NSObject, XCTestObservation {
}
// Test Observation
// Example LoggingObserver

class LoggingObserver: NSObject, XCTestCaseObservation {

    override init() {
        super.init()
        let sharedCenter = XCTestObservationCenter.shared()
        sharedCenter.addTestObserver(self)
    }

}
class Logger: NSObject, XCTestObservation {

    override init() {
        super.init()
        let sharedCenter = XCTestObservationCenter.shared()
        sharedCenter.addTestObserver(self)
    }

    internal func testBundleWillStart(_ testBundle: Bundle) {
        print("Lights down, show is about to start")
    }
}
class LoggingObserver: NSObject, XCTestObservation {

    override init() {
        super.init()
        let sharedCenter = XCTestObservationCenter.shared()
        sharedCenter.addTestObserver(self)
    }

    internal func testBundleWillStart(_ testBundle: Bundle) {
        print("Lights down, show is about to start")
    }

    internal func testCase(_ testCase: AnyObject!, didFailWithDescription description: AnyObject!, inFile filePath: AnyObject!, atLine lineNumber: AnyObject!)
    print("That sounds off key...")
}
// Test Observation
// Example LoggingObserver

class LoggingObserver: NSObject, XCTestCase {

    override init() {
        super.init()
        let sharedCenter = XCTestCaseObservationCenter.shared()
        sharedCenter.addTestObserver(self)
    }

    internal func testBundleWillStart(_ testBundle: Bundle) {
        print("Lights down, show is about to start")
    }

    internal func testCase(_ testCase: AnyObject!, didFailWithDescription description: AnyObject!, inFile filePath: AnyObject!, atLine lineNumber: AnyObject!) {
        print("That sounds off key...")
    }

    internal func testBundleDidFinish(_ testBundle: Bundle) {
        print("Lights up, don’t forget your coat!")
    }
}
**Test Observation**

Use NSPrincipalClass

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information Property List</td>
<td>Dictionary</td>
<td>(10 items)</td>
</tr>
<tr>
<td>Principal class</td>
<td>String</td>
<td>MyAppUITests.LoggingExample</td>
</tr>
<tr>
<td>Localization native development region</td>
<td>String</td>
<td>en</td>
</tr>
<tr>
<td>Executable file</td>
<td>String</td>
<td>$(EXECUTABLE_NAME)</td>
</tr>
<tr>
<td>Bundle identifier</td>
<td>String</td>
<td>$(PRODUCT_BUNDLE_IDENTIFIER)</td>
</tr>
<tr>
<td>InfoDictionary version</td>
<td>String</td>
<td>6.0</td>
</tr>
<tr>
<td>Bundle name</td>
<td>String</td>
<td>$(PRODUCT_NAME)</td>
</tr>
<tr>
<td>Bundle OS Type code</td>
<td>String</td>
<td>BNDL</td>
</tr>
<tr>
<td>Bundle versions string, short</td>
<td>String</td>
<td>1.0</td>
</tr>
<tr>
<td>Bundle creator OS Type code</td>
<td>String</td>
<td>???</td>
</tr>
<tr>
<td>Bundle version</td>
<td>String</td>
<td>1</td>
</tr>
</tbody>
</table>
Test Observation
Use NSPrincipalClass

Test Bundle Info.plist
Test Observation
Use NSPrincipalClass

Test Bundle Info.plist
Test specific instantiation
Testing Concepts

Summary

Compilation  Hosting  Observation
Crash Log Gathering
Crashes during testing
Crash Log Gathering
Crashes during testing
Crash Log Gathering

Crashes during testing

Test host or target application may crash
Crash Log Gathering
Crashes during testing

Test host or target application may crash
Xcode completes test suite
Crash Log Gathering

Crashes during testing

Test host or target application may crash
Xcode completes test suite
But how to resolve the crash?
Crash Log Gathering
Improved test reports
Crash Log Gathering

Improved test reports
Crash Log Gathering
Improved test reports
For Unit and UI tests
Crash Log Gathering

Improved test reports

For Unit and UI tests

Local and Xcode Server crash logs
Crash Log Gathering

Improved test reports

For Unit and UI tests

Local and Xcode Server crash logs

Logs included in test report
Crash Log Gathering

Improved test reports

For Unit and UI tests
Local and Xcode Server crash logs
Logs included in test report
View crashes in Xcode’s Debug Navigator
Demo
Crash Log Gathering
Crash Log Gathering
Improved test reports
Crash Log Gathering
Improved test reports

Crash logs gathered in test reports
Crash Log Gathering
Improved test reports

Crash logs gathered in test reports
View crashes in Xcode’s Debug Navigator
Xcode Server
Continuous Integration in Xcode

Eric Dudiak
Xcode Engineer
Xcode Server
Overview
Xcode Server

Overview

Custom environment variables
Xcode Server
Overview

Custom environment variables
Advanced trigger editing
Xcode Server

Overview

Custom environment variables
Advanced trigger editing
Issue tracking and blame
Xcode Server

Overview

- Custom environment variables
- Advanced trigger editing
- Issue tracking and blame
- Upgrade integrations
Xcode Server

Overview

- Custom environment variables
- Advanced trigger editing
- Issue tracking and blame
- Upgrade integrations
- Configurable integration user
Custom Environment Variables
New in Xcode 7.3
Advanced Triggers
Integration scripts
Advanced Triggers

Email notifications

Edit Harmony Bot:

- **Post-Integration Scripts**
  - Set beta icon

- **Email Notifications**
  - New issue email
  - Weekly report

**Schedule**
- Weekly on Friday at 17:00

**From**
- server@bots.example.com

**Reply To**
- myteam@example.com

**Cc**
- managers@example.com

**To**
- myteam@example.com

- Email committers

**Include**
- Issues
- Commits
- Server information

Filter Recipients...
Advanced Triggers

Email notifications
Advanced Triggers

Email notifications

Edit Harmony Bot:

- Post-Integration Scripts
  - Set beta icon
  - Email Notifications
    - New issue email
    - Weekly report

Schedule:
- Weekly on Friday at 17:00

From: server@bots.example.com
Reply To: myteam@example.com
Cc: managers@example.com
To: myteam@example.com

Include:
- Issues
- Commits
- Server Information

Options:
- Email committers
- Filter Recipients

Buttons:
- Cancel
- Done
Advanced Triggers

Email notifications
Advanced Triggers

Email notifications
Advanced Triggers

Email notifications

Choose repositories for which emails will be sent:
- https://gitbucket.example.com/scm/Harmony.git
- https://gitbucket.example.com/scm/EXFramework.git

Enter domain names for which emails will be sent:
- groups.example.com

Notify On:
- Build errors
- Build warnings
- Static analysis issues
We all have issues
Issue Tracking and Blame

We all have issues

Nobody is perfect
Issue Tracking and Blame

We all have issues

Nobody is perfect

• Tests will fail
Issue Tracking and Blame

We all have issues

Nobody is perfect

• Tests will fail
• Errors will come up
We all have issues

Nobody is perfect

- Tests will fail
- Errors will come up
- Xcode will blame you
We all have issues

Nobody is perfect

• Tests will fail
• Errors will come up
• Xcode will blame you

Even if you are perfect
Issue Tracking and Blame

We all have issues

Nobody is perfect

- Tests will fail
- Errors will come up
- Xcode will blame you

Even if you are perfect

- SDKs change
Issue Tracking and Blame

We all have issues

Nobody is perfect
• Tests will fail
• Errors will come up
• Xcode will blame you

Even if you are perfect
• SDKs change
• Language improvements
We all have issues

Nobody is perfect
• Tests will fail
• Errors will come up
• Xcode will blame you

Even if you are perfect
• SDKs change
• Language improvements
• Smarter tools
Identifying issue owners

Bobert the Bot
You introduced 1 issue on Harmony Bot
To: John Appleseed
Reply-To: My Team

You introduced 1 issue on Harmony Bot
Integration #2 ran on Tuesday June 14th on teamsserver.example.com and detected 1 warning on “Harmony Bcd”. This issue introduced by 1 commit you made to the Harmony xcodeproj repository (1e6f973). This bot is configured to notify committers they introduce new issues.

Download Integration Logs | Open in Xcode

After fixing this issue, you should trigger a manual integration as this bot is not configured to run on a schedule.

Issues (1)
Warning in Harmony/ViewController.swift:22
User-Defined Issue: Reminder to revisit this code tomorrow
Introduced by John Appleseed (1e6f973)

Commits (1)
https://gitbucket.example.com/scm/Harmony.git (master)
1e6f973 1 Warning John Appleseed - Jun 14, 2016 10:45 PM
Adding a reminder for later.

Server information
Hostname: teamsserver.example.com
Software:
OS X 10.12
Xcode 8.0, Server 5.1.50
Identifying issue owners

Lets you know when it’s time to act
Issue Tracking and Blame
Identifying issue owners

Lets you know when it’s time to act
You broke it
Issue Tracking and Blame
Identifying issue owners

Lets you know when it’s time to act
You broke it
- You introduced an issue
Identifying issue owners

Lets you know when it’s time to act

You broke it

• You introduced an issue
• Issue is on or near line you modified
Issue Tracking and Blame
Identifying issue owners

Let's you know when it's time to act

You broke it

- You introduced an issue
- Issue is on or near line you modified
- You are the only committer
Issue Tracking and Blame
Identifying issue owners

Lets you know when it’s time to act

You broke it
• You introduced an issue
• Issue is on or near line you modified
• You are the only committer

You can help fix it
Identifying issue owners

Lets you know when it’s time to act

You broke it

• You introduced an issue
• Issue is on or near line you modified
• You are the only committer

You can help fix it

• You probably know how to fix an issue
Identifying issue owners

Lets you know when it’s time to act
You broke it
• You introduced an issue
• Issue is on or near line you modified
• You are the only committer
You can help fix it
• You probably know how to fix an issue
• You seem to know about the area involved
Identifying issue owners

Lets you know when it’s time to act
You broke it
• You introduced an issue
• Issue is on or near line you modified
• You are the only committer
You can help fix it
• You probably know how to fix an issue
• You seem to know about the area involved
• Fuzzy matching
Issue Tracking and Blame

Bot configuration changes

Bobert the Bot
Integration #5 of Harmony Bot Failed with 1 Test Failure
To: John Appleseed
Reply-To: My Team

Integration #5 of Harmony Bot Failed with 1 Test Failure

The total number of passing tests in this integration was 14. There was 1 commit.

Summary
Bot Configuration Changes
Testing enabled

Integration Results
15 tests
14 passing tests
1 test failure (+1)

Committers
John Appleseed

Open Issues Introduced in This Integration
Test Failure in Harmony/HarmonyTests/HarmonyTests.swift:41
Assertions: XCTAssertEqual failed. ("Optional(-1.49)") is not equal to ("Optional(-1.49)")
Introduced by configuration change

Commits (1)
https://gitbucket.example.com/scm/Harmony.git (master)
1d6f573 John Appleseed - Jun 14, 2016 10:48 PM
Editing a test

Server information
Issue Tracking and Blame

Bot configuration changes

Tracks changes to your bot configuration
Issue Tracking and Blame

Bot configuration changes

Tracks changes to your bot configuration
Attributes issues to changes where possible
Issue Tracking and Blame

Bot configuration changes

Tracks changes to your bot configuration
Attributes issues to changes where possible
Included in email reports
Upgrade Integrations
When it’s not your fault
Upgrade Integrations
When it’s not your fault

Re-integrates your project
Upgrade Integrations
When it’s not your fault

Re-integrates your project
Same revision as the last integration
Upgrade Integrations
When it’s not your fault

Re-integrates your project
Same revision as the last integration
Any new issues are due to the upgrade
Upgrade Integrations
When it’s not your fault

Re-integrates your project
Same revision as the last integration
Any new issues are due to the upgrade
Prevents blaming you for broken builds
Configurable Integration User
Configurable Integration User

Improved visibility into your integrations
Configurable Integration User

Improved visibility into your integrations
Allows customization
Configurable Integration User

Improved visibility into your integrations
Allows customization
You own and manage how your integrations run
Configurable Integration User

Improved visibility into your integrations
Allows customization
You own and manage how your integrations run
Can be any user on the system
Configurable Integration User

Improved visibility into your integrations
Allows customization
You own and manage how your integrations run
Can be any user on the system
Runs as a menu extra
Custom Integration User

Improved visibility into your integrations

Allows customization

You own and manage how your integrations run

Can be any user on the system

Runs as a menu extra
Custom Integration User

Improved visibility into your integrations

Allows customization

You own and manage how your integrations run

Can be any user on the system

Runs as a menu extra
Custom Integration User

Improved visibility into your integrations

Allows customization

You own and manage how your integrations run

Can be any user on the system

Runs as a menu extra
Custom Integration User

Improved visibility into your integrations

Allows customization

You own and manage how your integrations run

Can be any user on the system

Runs as a menu extra
Choose Xcode to begin

Use Xcode Server to continually build and test your team's apps.
Download the latest version of Xcode

Choose Xcode...
You must log in as Customized Integration User to run integrations.

Integrations will not start until you log in. If you use fast user switching to return to this user, integrations will continue to run in the background.

Permissions: Logged in users can create bots, anyone can view bots

Build using: Xcode 8.0 (8A147)

Run tests as: Customized Integration User (Logged Out)

Developer Teams: No Teams

Development Devices: Name Type Status

Add a developer team to test on iOS devices
Custom Integration User

Improved visibility into your integrations

Allows customization

You own and manage how your integrations run

Can be any user on the system

Runs as a menu extra

Xcode

Settings  Repositories

Status: Available on your local network at Example-Server.local
Learn about configuring devices for this service.

Permissions: Logged in users can create bots, anyone can view bots

Edit Permissions...

Build using: Xcode 8.0 (8A147)

Choose Xcode...

Run tests as: Customized Integration User (Logged In)

Switch User...

Developer Teams: No Teams

Add Team...

Development Devices:

Name  Type  Status

Add a developer team to test on iOS devices

View bots

Search
Demo
Xcode Server
Configurable Integration User
Configurable Integration User

Improved visibility
Configurable Integration User

Improved visibility
Customized simulators
Configurable Integration User

Improved visibility
Customized simulators
Normal macOS user
Configurable Integration User

Improved visibility
Customized simulators
Normal macOS user
Runs as a menu extra
Configurable Integration User
Best practices
Configurable Integration User

Best practices

Dedicate a new user
Configurable Integration User

Best practices

Dedicate a new user

Avoid using administrator accounts
Configurable Integration User

Best practices

Dedicate a new user
Avoid using administrator accounts
Stay logged in with Fast User Switching
Configurable Integration User

Best practices

Dedicate a new user
Avoid using administrator accounts
Stay logged in with Fast User Switching
Disable screen lock
Configurable Integration User

Best practices

Dedicate a new user
Avoid using administrator accounts
Stay logged in with Fast User Switching
Disable screen lock
Customize for your needs
Configurable Integration User

Best practices

Dedicate a new user
Avoid using administrator accounts
Stay logged in with Fast User Switching
Disable screen lock
Customize for your needs
  • Simulators
Configurable Integration User

Best practices

Dedicate a new user
Avoid using administrator accounts
Stay logged in with Fast User Switching
Disable screen lock
Customize for your needs
  • Simulators
  • Networking (VPN and proxies)
Best practices

Dedicate a new user
Avoid using administrator accounts
Stay logged in with Fast User Switching
Disable screen lock
Customize for your needs
• Simulators
• Networking (VPN and proxies)
• User data and settings
Configurable Integration User

Best practices

Dedicate a new user
Avoid using administrator accounts
Stay logged in with Fast User Switching
Disable screen lock
Customize for your needs
  • Simulators
  • Networking (VPN and proxies)
  • User data and settings
  • Advanced provisioning
xcodebuild
Building block for Continuous Integration Systems
xcodebuild

Test action

```
xcodebuild test -workspace <path>
   -scheme <name>
   -destination <specifier>
```
xcodebuild
Test action

xcodebuild test -workspace <path>
    -scheme <name>
    -destination <specifier>

Builds sources
xcodebuild

Test action

xcodebuild **test** -workspace `<path>`
   -scheme `<name>`
   -destination `<specifier>`

Builds sources
Installs app on destination if needed
xcodebuild
Test action

xcodebuild test -workspace <path>
-scheme <name>
-destination <specifier>

Builds sources
Installs app on destination if needed
Runs tests
xcodebuild
Test action

xcodebuild test -workspace <path>
   -scheme <name>
   -destination <specifier>

Builds sources
Installs app on destination if needed
Runs tests
Reports results in console
Test Options
Customizing tests

xcodebuild test -workspace <path>
-scheme <name>
-destination <specifier>
-only-testing:TestBundleA/TestSuiteA/TestCaseA
-only-testing:TestBundleB/TestSuiteB
-only-testing:TestBundleC
Test Options
Customizing tests

```
xcodebuild test -workspace <path>
  -scheme <name>
  -destination <specifier>
  -only-testing:TestBundleA/TestSuiteA/TestCaseA
  -only-testing:TestBundleB/TestSuiteB
  -only-testing:TestBundleC

xcodebuild test -scheme <name>
  -skip-testing:TestBundleD/TestSuiteD/TestCaseD
```
Additional Test Actions
Separating building from testing
Additional Test Actions
Separating building from testing

- Build for Testing
- Test without Building
Building for Testing

Only building

```
xcodebuild build-for-testing -workspace <path>
    -scheme <name>
    -destination <specifier>
```
Building for Testing
Only building

```
xcodebuild build-for-testing -workspace <path>
  -scheme <name>
  -destination <specifier>
```

Builds sources for testing
Building for Testing

Only building

```
xcodebuild build-for-testing -workspace <path>
   -scheme <name>
   -destination <specifier>
```

Builds sources for testing

Outputs products in Derived Data
Building for Testing

Only building

```
xcodebuild build-for-testing -workspace <path>
   -scheme <name>
   -destination <specifier>
```

Builds sources for testing
Outputs products in Derived Data
Generates an xctestr run file
Test Without Building
Only testing

xcodebuild test-without-building -workspace <path>
  -scheme <name>
  -destination <specifier>
Test Without Building

Only testing

```
xcodebuild test-without-building -workspace <path>
  -scheme <name>
  -destination <specifier>
```

Finds binary products in Derived Data
Test Without Building
Only testing

```
xcodebuild test-without-building -workspace <path>
    -scheme <name>
    -destination <specifier>
```

Finds binary products in Derived Data
Installs if necessary
Test Without Building

Only testing

```bash
xcodebuild test-without-building -workspace <path>
  -scheme <name>
  -destination <specifier>
```

Finds binary products in Derived Data
Installs if necessary
Runs tests and report results as before
Test Without Building

Only testing

$xcodebuild test-without-building -xctestrun <path>
-destination <specifier>$
Test Without Building

Only testing

xcodebuild test-without-building -xctestr <path> -destination <specifier>

Ingests the xctestr file
Test Without Building

Only testing

```
xcodebuild test-without-building -xctestr <path>
   -destination <specifier>
```

Ingests the xctestr file

Finds binary products relative to xctestr file
Test Without Building

Only testing

```
xcodebuild test-without-building -xctestrun <path>
   -destination <specifier>
```

Ingests the xctestrun file

Finds binary products relative to xctestrun file

Runs tests, reports results as before
xcodebuild
Distributed testing

Build Machine

Test Machine #1

Test Machine #2
Distributed testing

Build Machine

Test Machine #1

Test Machine #2
Distributed testing

Build Machine Test Machine #2

Test Machine #1

xcodebuild build-for-testing
Distributed testing

Build Machine

Test Machine #1

Test Machine #2

xcodebuild build-for-testing
**xcodebuild**

Distributed testing

- **Build Machine**: xcodebuild `build-for-testing`
- **Test Machine #1**: xcodebuild `test-without-building`
- **Test Machine #2**:
Distributed testing

xcodebuild build-for-testing

Build Machine

Test Machine #1

xcodebuild test-without-building

Test Machine #2

xcodebuild test-without-building
xctestrrun
Test manifest
xcteestrun
Test manifest
xctestrun
Test manifest

Tests to run or skip
xcteestrun
Test manifest

Tests to run or skip

Environment variables
xcteestrn
Test manifest

Tests to run or skip
Environment variables
Command-line arguments
xcteestrunt
Test manifest

Tests to run or skip
Environment variables
Command-line arguments
Documentation in man pages
xctestrun
Test manifest

Tests to run or skip
Environment variables
Command-line arguments
Documentation in man pages

man xcodebuild.xctestrun
Summary

Testing Concepts

Xcode
- Crash Log Gathering

Xcode Server
- Advanced Triggers
- Issue Tracking and Blame
- Configurable Integration User

xcodebuild
More Information

https://developer.apple.com/wwdc16/409
### Labs

<table>
<thead>
<tr>
<th>Xcode Open Hours</th>
<th>Developer Tools Lab C</th>
<th>Thursday 9:00AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xcode Open Hours</td>
<td>Developer Tools Lab B</td>
<td>Friday 9:00AM</td>
</tr>
</tbody>
</table>