

# Becoming a Pro

## IN Mobile Applications Testing



# HOMWORK

## LEARN HOW TO COLLECT LOGS



# Mobile APPS: *Collecting LOGS: .IPA*



How to do it through Xcode on MAC ?

1. Install XCode

2. Connect your iPhone to the Mac

3. Select Trust this computer on the iPhone pop-up request

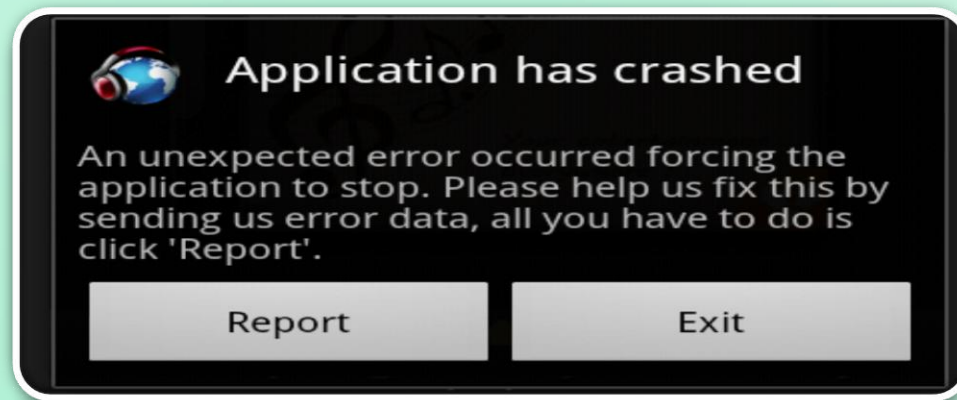
4. Start xCode (Menu) → Window → Devices (Select your iPhone and press the arrow button in the bottom right)

5. Reproduce the problem

6. Press the arrow button and download the logs

# How can I debug a deployed app without Xcode debugger ?

- ❖ A: Once you have deployed your app, either through the *App Store* or as an *Ad Hoc* or *Enterprise* build, you won't be able to attach *Xcode's* debugger to it.
- ❖ To debug problems, you need to analyze *Crash Logs* and *Console* output from the device.



# Getting Crash Logs and Console Output

## Getting Crash Logs Directly From a Device

### Without Xcode

Users can retrieve crash reports from their device and send them to you via email by following these instructions.

(It is not possible to get device console logs directly from a device)

- ❖ Open Settings app
- ❖ Go to Privacy, then Diagnostics & Usage
- ❖ Select Diagnostics & Usage Data
- ❖ Locate the log for the crashed app. The logs will be named in the format: *<AppName>\_<DateTime>\_<DeviceName>*
- ❖ Select the desired log. Then, using the text selection UI select the entire text of the log. Once the text is selected, tap Copy
- ❖ Paste the copied text to Mail and send to an email address as desired

# Mobile APPS: **Distribution/Installation: .IPA**



What is iTunes?

iTunes is a media player, media library, online radio broadcaster, and mobile device management application developed by Apple Inc.

It is used to play, download, and organize digital downloads of music and video (as well as other types of media available on the iTunes Store) on personal computers running the macOS and Microsoft Windows operating systems.

What is iTunes?



# Mobile APPS: **Distribution/Installation: .IPA**

*Drag-and-drop IPA file into 'Apps' tab of iTunes BEFORE you connect the device*



*Connect your device*

*Select your device on iTunes*

*Select 'Apps' tab*

*Search app that you want to install*

*Click on 'Install' button. This will change to 'Will Install'*

*Click on 'Apply' button on right corner*

# ***LOGS: Collecting from iTunes***

- ❖ Sync your device with iTunes on your desktop.
- ❖ After syncing, look for crash logs in the correct directory.

***The Following few slides will give an instructions . Lets start !***





# ***LOGS: Collecting from iTunes***

## ❑ Mac OS X:

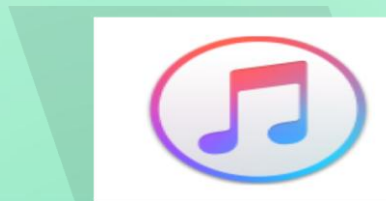
- ❖ Open Finder (found in the Dock)
- ❖ Click on the 'Go' menu at the top of your screen, and select 'Go to Folder'
- ❖ Type (or paste): `~/Library/Logs/CrashReporter/MobileDevice/<DEVICE_NAME>`
- ❖ Open the folder with the same 'name' as your device.  
(Note: your device name appears in iTunes on the left side, under 'Devices').
- ❖ Open the folder called 'Retired'
- ❖ You will see at least one item starting with 'ReadItLaterPro'.



# ***LOGS: Collecting from iTunes***

## ❑ Windows Vista or 7:

- ❖ Open any Windows Explorer Window (My Computer, My Documents, etc.)
- ❖ Enter %appdata%, and press enter → Navigate to Roaming  
*C:\Users\<USERNAME>\AppData\Roaming\Apple*
- ❖ *Computer\Logs\CrashReporter\MobileDevice\<DEVICE\_NAME>*
- ❖ (Note: your device name appears in iTunes on the left side, under 'Devices')
- ❖ You will see at least one item starting with 'ReadItLaterPro'.



# ***LOGS: Syncing your device with iTunes***

## ❑ Windows XP:

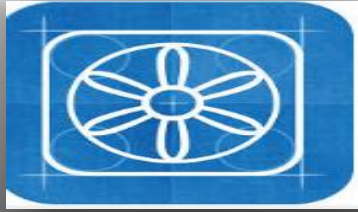
- ❖ Locate your Application Data folder.
- ❖ Navigate to Apple computer *C:\Documents and Settings\<USERNAME>\Application Data\Apple*
- ❖ *Computer\Logs\CrashReporter\MobileDevice\<DEVICE\_NAME>*
- ❖ (Note: your device name appears in iTunes on the left side, under 'Devices')
- ❖ You will see at least one item starting with 'ReadItLaterPro'.



# ***LOGS: Collecting from your Device***

- To obtain iOS crash logs, please select your device and go to:
- *Settings > Privacy > Diagnostics & Usage (for iOS 8 or newer)*
- *Settings > General > About > Diagnostics & Usage (for iOS 7 or older)*
  
- ❖ Select a ***Chrome*** crash from the list.  
This will start with “***Chrome\_***” and contain the timestamp of the crash.
- ❖ Tap on the crash and you will see a text field with a crash log.  
Long press to *Select All* and then *Copy* the crash text.
- ❖ Paste it into something you can get off of your device (for example, an email to yourself).

# Mobile APPS: **Distribution/Installation: .IPA**



What is TestFlight?

TestFlight is an online service for over-the-air installation and testing of mobile applications, currently owned by Apple Inc and only offered to developers within the iOSDeveloper Program

Developers signed up with the service to distribute applications to internal or external beta testers, who could subsequently send feedback about the application to developers

The TestFlight SDK additionally allows developers to receive remote logs, crash reports and tester feedback.

**HOMEWORK : READ AN ENTIRE ARTICLE ABOUT TestFLIGHT**

**<https://www.raywenderlich.com/48750/testflight-sdk-tutorial>**



# Overview: Mobile APPS

➤ Categories

➤ Types

➤ Distribution/Installation/Logs

➤ Mobile Test Industry Standards

➤ Remote Device Access (RDA)

➤ Emulators

➤ Simulators

➤ Troubleshooting Guide

➤ App Risk Analysis

# Mobile Test Industry Standards :

## Testing Strategies for Mobile Apps

### Challenges

Device fragmentation

In-house vs. outsourced testing

Availability of mobile testing tools

Application Lifecycle Testing

- Like any desktop or web application testing, mobile application testing is also focused on the quality and performance of the final product.
- However, mobile app testing becomes far more challenging because of the following key factors



# Mobile Test Industry Standards :

## Testing Strategies for Mobile Apps

Device Model

OS Version

Screen Resolution

Form Factor

Emulators vs. Physical  
Devices





# Mobile Test Industry Standards :

## Testing Strategies for Mobile Apps

**Network density**

**How the app behaves on specific devices**

**How real-world users interact with the app**

**Different battery states on the devices**

**Multiple networks (Wi-Fi, 4G, 3G, etc.)**

**Beta Testing of your Mobile App**



# Mobile Test Industry Standards :

## Testing Strategies for Mobile Apps

applications can be deployed, tested, and managed

saves businesses from setting up on-premise test environments

capability to support complex apps

provides real-time testing results

### Mobile App Testing on Cloud



# Mobile Test Industry Standards :

## Testing Strategies for Mobile Apps

testing mobile apps in real network environments

network simulation tools are available

test mobile apps in various network speeds, bandwidths variations

testing the app in a full internet connectivity scenario and other factors



# Mobile Test Industry Standards :

## Testing Strategies for Mobile Apps

Automated testing is highly effective in consistently repeating a test procedure in regression testing as well as testing during the development stages.

However, test automation requires significant amount of initial investment.



# Mobile Test Industry Standards :

## Testing Strategies for Mobile Apps

# Types of Mobile App Testing



# Mobile Test Industry Standards :

## Testing Strategies for Mobile Apps

### FUNCTIONAL TEST

Verifying that all documented requirements are implemented.

Verifying that all features work as expected.

Validating texts, logos, images, text captions and other UI elements.

Validating localization and globalization.

Evaluating ease of navigation and screen transitions.

Examining response speed.

Evaluating the intuitiveness of the touch interface.

# Mobile Test Industry Standards :

## Testing Strategies for Mobile Apps

### PERFORMANCE TEST

Performance with low battery power

Performance while network out of coverage area

Performance during poor bandwidth

Performance while changing internet connection mode

Performance while transferring heavy file

Testing from Application's server and client side

# **Mobile Test Industry Standards :**

## **Testing Strategies for Mobile Apps**

### **Memory Leakage TEST**

**Verifying if program runs for an extended time and consumes additional memory**

**Verifying if memory is allocated frequently for one-time tasks**

**Verifying where the program can request memory — such as shared memory that is not released**

**Verifying where memory is very limited, such as in an embedded system or portable device**

**Verifying where the leak occurs within the operating system or memory manager**

**Verifying when a system device driver causes the leak**



# **Mobile Test Industry Standards :**

## **Testing Strategies for Mobile Apps**

### **INTERRUPT TEST**

**Battery low**

**Battery full- when charging**

**Incoming phone call**

**Incoming SMS**

**Incoming Alert from another mobile application**

**Plugged in for charging**

**Plugged out from charging**

**Device shut off**

**Application Update reminders**

**Alarm**

**Network connection loss**

**Network connection restoration**