



---

## Running Calabash iOS Tests in Phresco

Version 3.2

February 2014

This document purports the execution steps involved in Calabash iOS tests.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

© Copyright Photon Interactive Pvt Ltd 2014.

All rights reserved.

All Photon product names are either trademarks or registered trademarks of Photon. Other company and product names mentioned herein may be trademarks of their respective owners.

## TABLE OF CONTENTS

1	About This Guide .....	3
2	Functional script structure .....	3
3	Develop scripts in Calabash .....	3
4	Pre-requisites for running test in Phresco .....	4
5	Executing test in Phresco .....	4

## 1 About This Guide

---

The purpose of this guide is to assist test engineers to develop iOS test scripts using Calabash with implementation in Ruby.

## 2 Functional script structure

---

The structure of the scripts should be as below

```
Parent Dir
/features/
Contains .feature file where acceptance criteria for scenarios are defined

-/features/step_definitions/
Contains Ruby methods for different scenarios as .rb file

-/features/support/
Contains Ruby implementation code for verifying scenarios as .rb files

-/features/.feature_file
Contains the test cases
```

## 3 Develop scripts in Calabash

---

Define acceptance criteria in features file, for all scenarios of your application. Develop Ruby code to verify the acceptance criteria's.

## 4 Pre-requisites for running test in Phresco

---

- Ruby 1.9.2 or above should be installed
- Run below commands from the path where xcode.proj is present
  - “gem install calabash-cucumber” to install calabash cucumber
  - “calabash-ios setup” to setup calabash ios
  - “calabash-ios gen” to generate the skeleton for features folder.
  - Build your project from Xcode.

## 5 Executing test in Phresco

---

1. Place the script in path /workspace/projects/project\_name/source
2. Execute test as “Calabash” against functional framework

