

CSCI 379: Software Design
Bucknell University
Computer Science Department
<http://www.eg.bucknell.edu/~cs379>

Logic Unit Testing

Table of Contents

- 1. Overview**
- 2. How to Set Up Unit Testing**
- 3. Writing Tests**

1. Overview

Logic unit testing won't run the application, but you can use it to test program logic. We will use it to test our application *model*.

2. How to Set Up Unit Testing

To set up logic unit testing for an application, follow these steps.

1. Add an iPhone OS unit-test bundle target to the project.
 - In the Groups & Files list, highlight **Targets**.
 - Select **Project** → **New Target...**
 - Make sure Cocoa Touch is selected on the left. Then click on Unit Test Bundle.
 - Press **Next**.
 - Use LogicTests as the Target Name.
 - Press **Finish**.
 - A Target “LogicTests” Info window will appear. Just close it. There's no need to enter anything.
2. Make the LogicTests target the active target.
 - Select **Project** → **Set Active Target** → **LogicTests**.
3. Add a group called Tests to the Group & Files list, and select that group in the list.
 - Highlight the project name in the Groups & Files list.
 - Select **Project** → **New Group** and name the group Tests.

4. Add a unit-test class to the unit-test-bundle target. Each unit-test class in the bundle makes up a test suite.
 - Click on the Tests group so it is selected.
 - Select **File** → **New File...** Make sure Cocoa Touch Class is selected on the left, click on Objective-C test case class and press **Next**.
 - Set the file name to `LogicTests.m`, check the box that says to also create `LogicTests.h`. In the Targets section, uncheck the box next to the project name, and check the box next to LogicTests.
 - Press **Finish**.
 - Add `LogicTests.h`, `LogicTests.m` and `Logic Tests-Info.plist` to your repository.
5. Change `LogicTests.h`.
 - Set the value of `USE_DEPENDENT_UNIT_TEST` to 0.
6. Change `LogicTests.m` by adding the following definition for `testFail` after the `#else`.

```
- (void) testFail {  
    STFail(@"Must fail to succeed.");  
}
```

7. Set the project active SDK to iPhone Simulator 3.0.
 - Select **Project** → **Set Active SDK** → **iPhone Simulator 3.0**.
8. Set the active target to LogicTests.

- Select **Project** → **Set Active Target** → **LogicTests**.
9. Select **Build** → **Build** (or click on the **Build** button) and observe the error.
 - The error is also visible in **Build** → **Build Results**
 10. Change the test to

```
- (void) testPass {
    STAssertTrue(TRUE, @"");
}
```
 11. Select **Build** → **Build** and observe that the tests pass.
 12. Select **Build and Go** to run the tests then the application.

3. Writing Tests

To test a particular class, you must do two things.

1. Import the header file for your class in **LogicTests.h**.
2. Add the implementation file for your class to the unit-test bundle target. To do this click on the disclosure triangle next to the Logic Tests target. You will see a folder called Compile Sources. Drag the implementation file you wish to test into this folder.

To discover the other testing macros that are available, see the *Unit-Test Result Macro Reference* which is available through the Xcode documentation. You can also find it in Appendix B of the **iPhone Development Guide**.