

< This slide left intentionally  
blank />

# ColdFusion + VBScript

Leveraging our favorite tools for  
cross\*-platform development

*Coming soon:* COBOL and Fortran connectors, dBase integration, and Rexx scripting!

*\* Cross: (adjective) annoyed: they seemed to be very cross about something.*

~~ColdFusion + VBScript~~

~~Leveraging our favorite tools for  
cross\* platform development~~

~~Coming soon: COBOL and Fortran connectors, dBase integration, and Rexx scripting!~~

No offense to these technologies!

# Getting to Know the XCTest Lifecycle


You write tests, right?

*Right...?*



What happens *around* our test  
methods and assertions?

## The Bare Essentials of an XCTestCase



```
class XCTestCase {  
    // What happens before this runs?  
    // After?  
    func testAddition() throws {  
        XCTAssertEqual(1 + 1, 2)  
    }  
}
```

# The Workhorses




```
override func setUp() {  
    super.setUp()  
    // Prepare test harness here...  
}  
  
override func tearDown() {  
    // Clean up test harness / environment here...  
    super.tearDown()  
}
```

- `setUp` runs before **each** test method
- `tearDown` runs after **each** test method
- You *can* make assertions in these, too



# The New(er/ish) Kids



```
override func setUpWithError() throws {  
    try super.setUpWithError()  
    // Prepare test harness here...  
}  
  
override func tearDownWithError() throws {  
    // Clean up test harness / environment here...  
  
    try super.tearDownWithError()  
}
```

- Since Xcode 11.4
- These do the same, but allow for actions that might throw an Error
- This results in cleaner Swift code when you need to work with errors here
- Xcode's template also shows the example test methods with **throws**

# These look similar, but...

```
override class func setUp() {  
    super.setUp()  
    // Prepare test harness here...  
}  
override class func tearDown() {  
    // Clean up test harness / environment here...  
    super.tearDown()  
}
```

- These run *once* for the **entire** test case
- You cannot have assertions in these

# Per-Test Method Clean Up



```
func testWithTeardownBlock() throws {  
    XCTAssertTrue(true)  
    addTeardownBlock {  
        // ... clean up ...  
    }  
}
```

- Teardown blocks are executed even when errors are thrown
- You cannot throw/rethrow errors that arise within their execution


# More Teardown Blocks



```
func testWithMultipleTeardownBlocks() throws {  
    XCTAssertTrue(true)  
    addTeardownBlock { /* ... */ }  
    XCTAssertFalse(false)  
    addTeardownBlock { /* ... */ }  
    XCTAssertTrue(true)  
    addTeardownBlock { /* ... */ }  
    XCTAssertFalse(false)  
}
```

- When there are multiple, they are executed LIFO
- You can make assertions within these blocks

# Other Hooks (via Inheritance)



```
override class var defaultTestSuite: XCTestSuite {  
    return super.defaultTestSuite  
}  
override func invokeTest() {  
    super.invokeTest()  
}  
override func record(_ issue: XCTIssue) {  
    super.record(issue)  
}
```

- `init(invocation: NSInvocation?)` is the designated initializer, but is unavailable for Swift
- Also, `deinit` does not get executed

Is that all?

# Another Set of Hooks

## XCTestObservation

For monitoring the overall test execution process  
Can be used for reporting or logging

# Getting Started

- Write a new class conforming to `XCTestObservation`
- Add an entry in the test target's `Info.plist` under key `NSPrincipalClass/Principal` **class** with a value of `$(PRODUCT_NAME).TheNameOfYourTestObserverType`
- Then have it add itself during initialization:

```
override init() {  
    super.init()  
  
    XCTestObservationCenter.shared.addTestObserver(self)  
}
```

- You can add multiple observers
- `removeTestObserver` also exists to remove an observer mid-session



# Keep your eye on the bundle



```
public func testBundleWillStart(_ testBundle: Bundle) { /* ... */ }  
public func testBundleDidFinish(_ testBundle: Bundle) { /* ... */ }
```



# Be on the case



```
public func testCaseWillStart(_ testCase: XCTestCase) { /* ... */ }

public func testCaseDidFinish(_ testCase: XCTestCase) { /* ... */ }

public func testCase(_ testCase: XCTestCase, didRecord issue: XCTIssue) { /* ... */ }

public func testCase(_ testCase: XCTestCase,
                    didFailWithDescription description: String,
                    inFile filePath: String?,
                    atLine lineNumber: Int) { /* ... */ }
```

# The Overall Sequences

# XCTestCase Sequence

## Example 1

```
● ● ●
i ==> defaultTestSuite
*   ==> class-level setUp
i ==> run
i ==> perform
i ==> invokeTest
*   ==> setUpWithError
*
*   ==> setUp
*   ----> TEST: testAddition
*   ==> tearDown
*   ==> tearDownWithError
i ==> run
i ==> perform
i ==> invokeTest
*   ==> setUpWithError
*   ==> setUp
*   ----> TEST: testThatFails
i ==> record issue: Assertion Failure at XCTestLifecycleTests.swift:159: XCTAssertEqual failed: ("true"
") is not equal to ("false") - Supposed to fail. Method: testThatFails()
*   ==> tearDown
*   ==> tearDownWithError
*   ==> class-level tearDown
```

# XCTestCase Sequence

## Example 2

```

● ● ●
i ==> defaultTestSuite
*   ==> class-level setUp
i ==> run
i ==> perform
i ==> invokeTest
*       ==> setUpWithError
*           ==> setUp
*               ----> TEST: testWithMultipleTearDownBlocks
*               ----> testWithMultipleTearDownBlocks THIRD teardown block
*               ----> testWithMultipleTearDownBlocks SECOND teardown block
*               ----> testWithMultipleTearDownBlocks FIRST teardown block
*           ==> tearDown
*       ==> tearDownWithError
* ==> class-level tearDown
```

# XCTestObservation Sequence



```
⌘ Observer: init
⌘ Observer: testBundleWillStart
⌘ Observer: testSuiteWillStart
⌘ Observer: testSuiteWillStart
⌘ Observer: testSuiteWillStart
⌘ Observer: testCaseWillStart
⌘ Observer: testCaseDidFinish
⌘ Observer: testCaseWillStart
⌘ Observer: test case did record issue with case (when test failures)
⌘ Observer: testCaseDidFinish
⌘ Observer: testCaseWillStart
⌘ Observer: testCaseDidFinish
⌘ Observer: testSuiteDidFinish
⌘ Observer: testSuiteDidFinish
⌘ Observer: testSuiteDidFinish
⌘ Observer: testBundleDidFinish
```

# Demo





Thanks!

Questions?

Kevin Munc \* [@muncman](#)