

JUnit

Dwight Deugo (dwight@espirity.com)
Nesa Matic (nesa@espirity.com)

Additional Contributors

- None as of August, 2004

Module Road Map

1. JUnit

Module Road Map

1. JUnit

- **What is JUnit?**
- **Where Does it Come From?**
- **Working with TestCases**
- **Working with TestSuites**
- **JUnit Window**

What is JUnit?

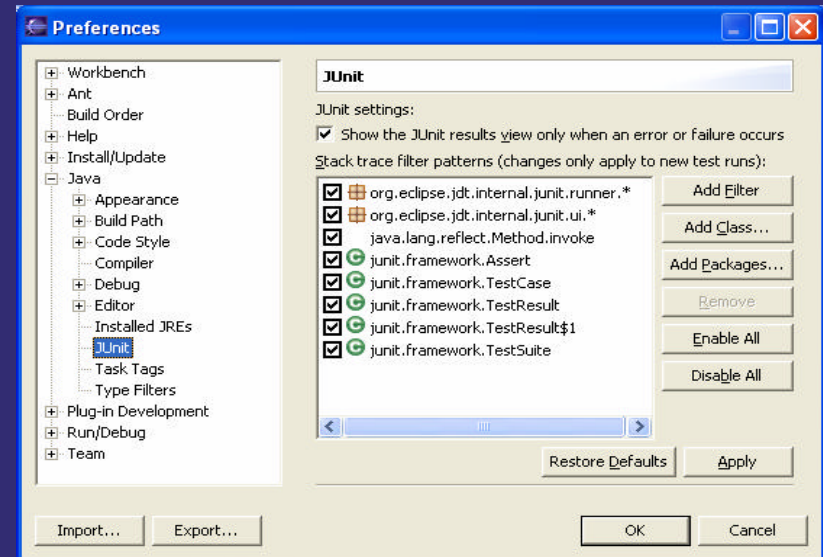
- Regression testing framework
- Written by Erich Gamma and Kent Beck
- Used for unit testing in Java
- Open Source
- Released under IBM's CPL

Where Does JUnit Come From?

- JUnit's web site: <http://junit.org/index.htm>
- Eclipse includes JUnit
 - Eclipse provides new GUI to run JUnit test cases and suites
- You can run your unit tests outside of Eclipse
 - If you wish using TestRunner
 - Using JUnit's Window

Eclipse JUnit Setup

- Eclipse preferences can be set in the JUnit Preferences window
- For the most part you can leave these alone
- Filters needed to identify packages, classes, or patterns that should not be shown in the stack trace of a test failure

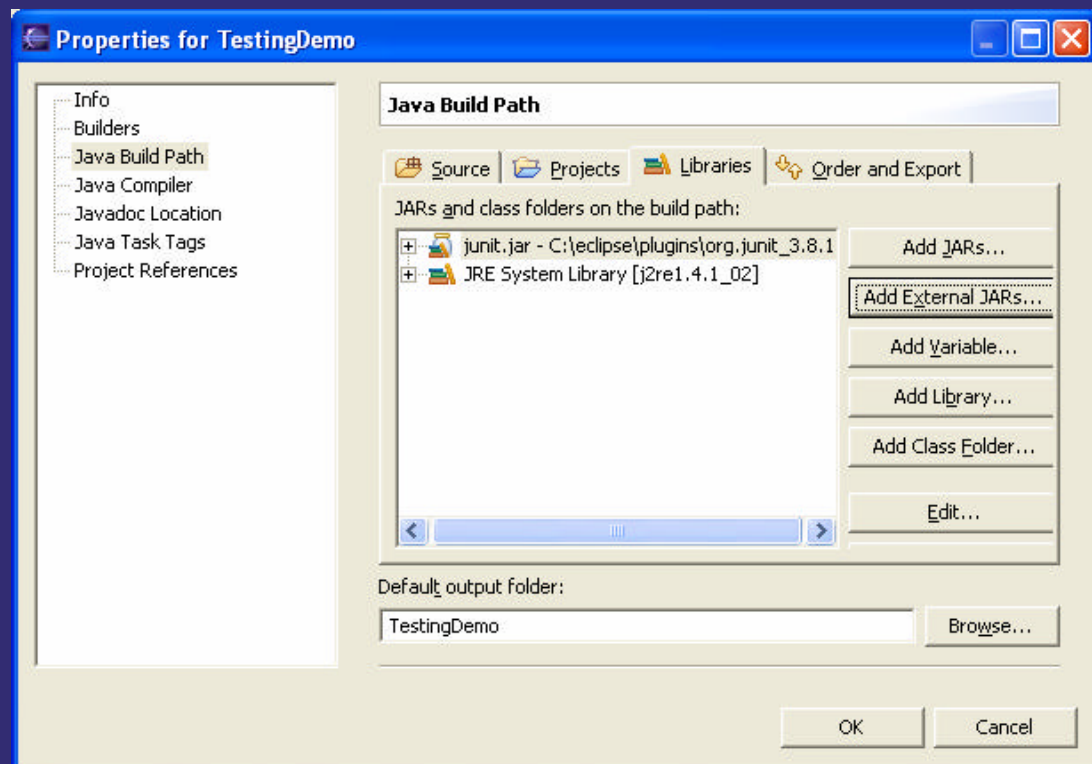


JUnit Test Cases

- Test case
 - Runs multiple tests
- Implemented a subclass of TestCase
- Define instance variables that store the state of the tests in the class
- Initialize TestCase by overriding setUp method
- Clean-up after test case is done by overriding tearDown method

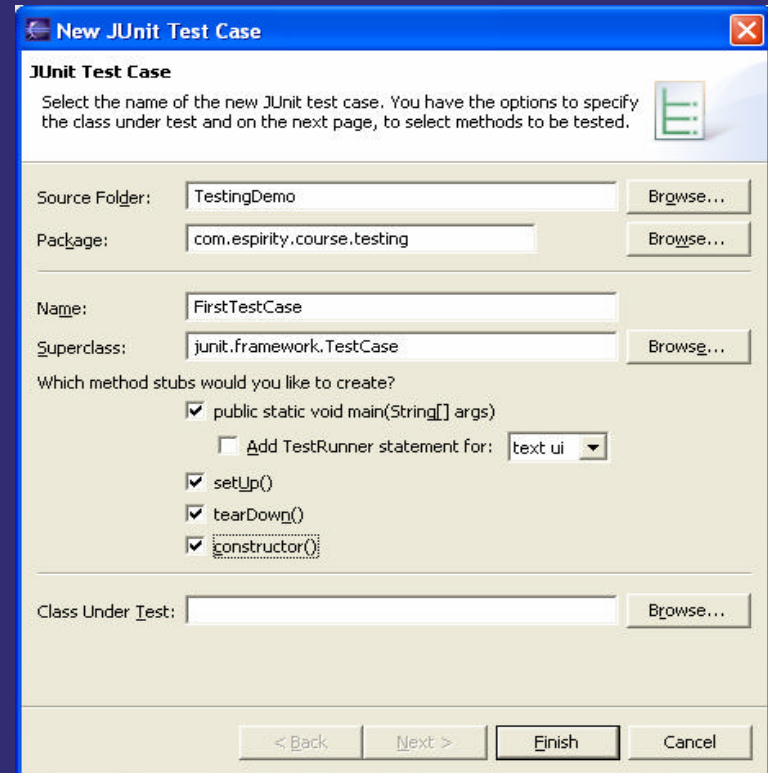
Creating TestCases in Eclipse...

- Create a new package to contain your test case classes
- Add the JUnit JAR file to the project's buildpath



...Creating TestCases in Eclipse

- Select your testing package
- From the context menu select New → JUnit Test Case
- In the next Window fill in the name of your test case
- This will create the corresponding class in your testing package



TestCase Template

```
package com.espirity.course.testing;
import junit.framework.TestCase;

public class FirstTestCase extends TestCase {

    public FirstTestCase(String arg0) {
        super(arg0);
    }

    public static void main(String[] args) {

    }

    protected void setUp() throws Exception {
        super.setUp();
    }

    protected void tearDown() throws Exception {
        super.tearDown();
    }
}
```

Adding Tests to TestCases

- Any method in a TestCase class is considered a test if it begins with the word test
 - You can write many tests (have many test methods)
- Each test method should use a variety of assert methods to test things about the state of their classes under tests
 - Assert methods are inherited

Assert Methods

- Assert methods include:
 - `assertEqual(x,y)`
 - `assertFalse(boolean)`
 - `assertTrue(boolean)`
 - `assertNull(object)`
 - `assertNotNull(object)`
 - `assertSame(firstObject, secondObject)`
 - `assertNotSame(firstObject, secondObject)`

Adding Two Tests to TestCase

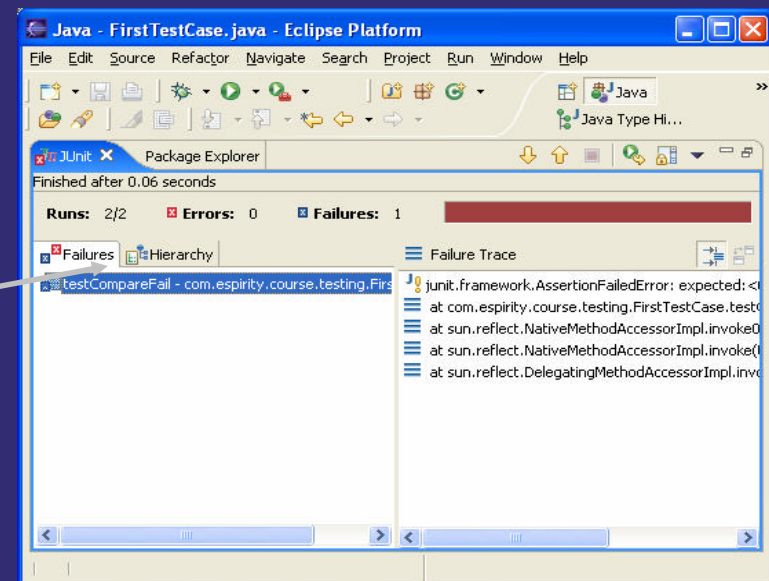
```
package testing;
import junit.framework.TestCase;

public class FirstTestCase extends TestCase {
    public FirstTestCase(String arg0) {
        super(arg0);
    }
    public static void main(String[] args) {}
    protected void setUp() throws Exception {
        super.setUp();
    }
    protected void tearDown() throws Exception {
        super.tearDown();
    }

    public void testCompareSucceed() {
        assertEquals(0, 0); //this assertion will succeed
    }
    public void testCompareFail() {
        assertEquals(0, 1); //this assertion will fail
    }
}
```

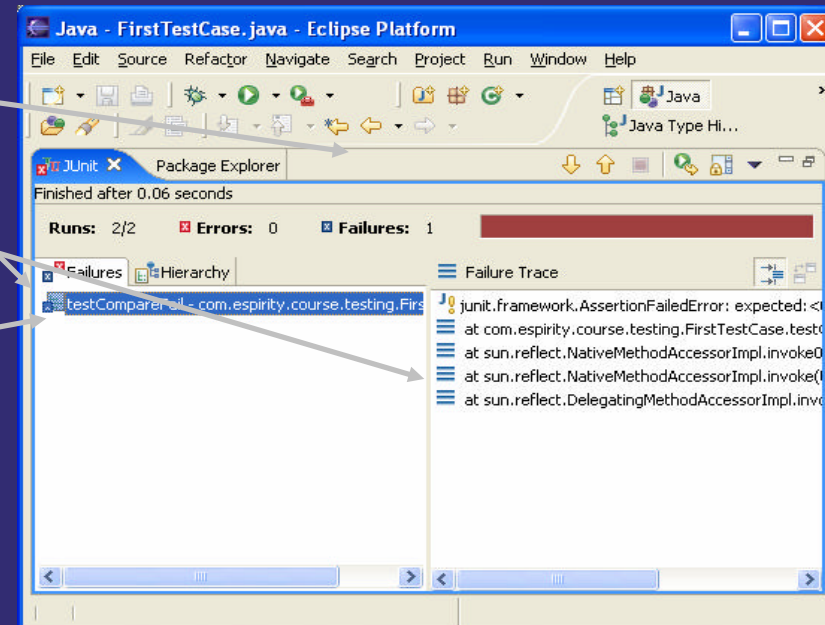
Running TestCase

- Select TestCase class
- From the Run menu select Run
→ Run As → JUnit Test
- This will run the tests in your TestCase class along with the setup and teardown methods
- You will then get a report in the JUnit Window



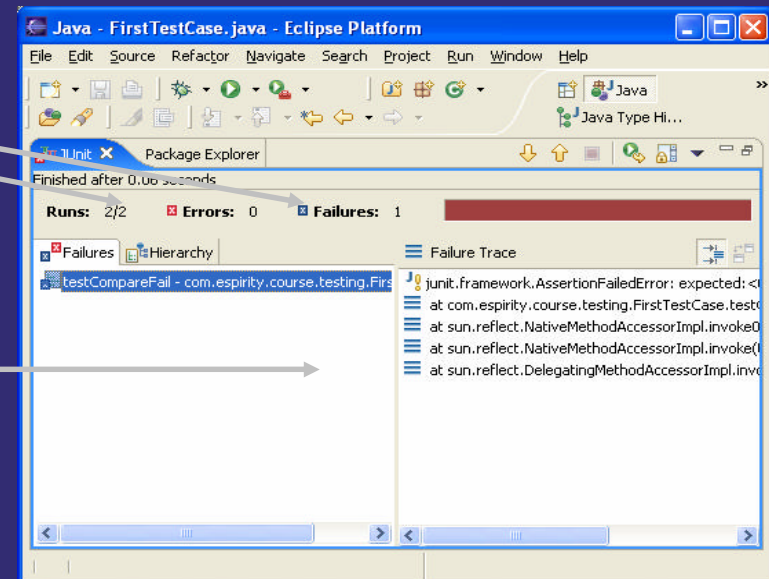
JUnit Window...

- Red indicates a test has failed
- You can see which test failed
- You can see the call trace leading to the failure
- If you wish to see the tests in TestCase click on the Hierarchy tab



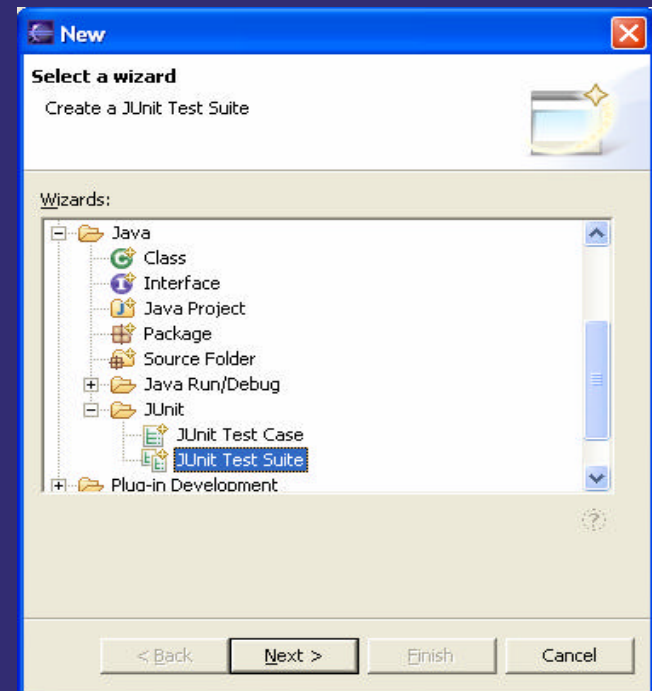
...JUnit Window

- You can see how many tests ran
- How many failures occurred
- You can see the details of the failure
- Errors occur when exceptions are thrown



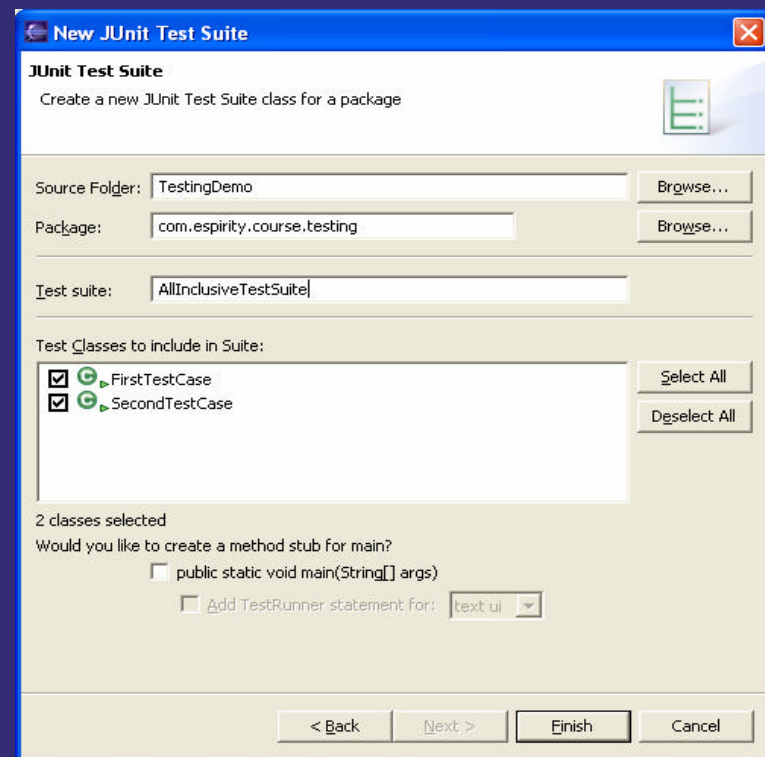
Creating JUnit TestSuite...

- Test Suite
 - Runs multiple test cases or suites
- Implemented as subclass of TestSuite
- To create a TestSuite
 - Select your testing package
 - From the context menu select New → Other... → Java → JUnit
 - Then from the Wizard select JUnit Test Suite



...Creating JUnit TestSuite

- Fill in the name of your TestSuite Class
- Select the TestCases to include in your TestSuite



TestSuite Template

```
package com.espirity.course.testing;

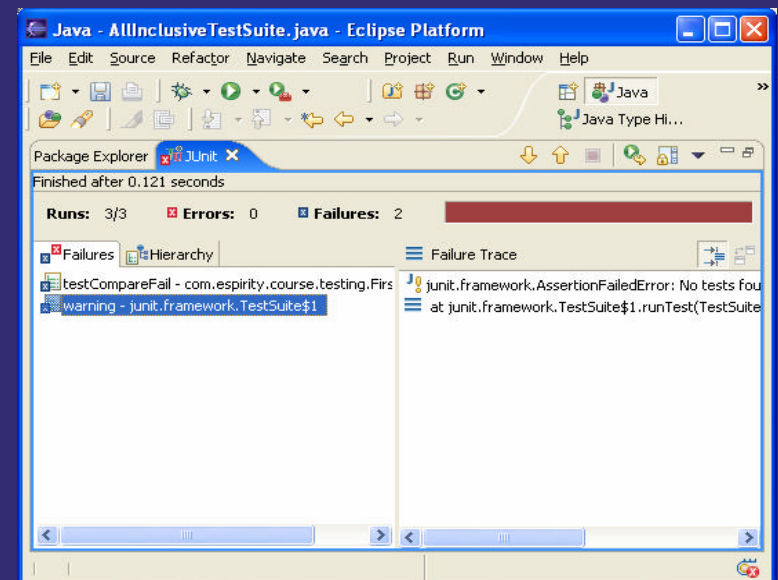
import junit.framework.Test;

public class AllInclusiveTestSuite {

    public static Test suite() {
        TestSuite suite =
            new TestSuite("Test for com.espirity.course.testing");
        //$JUnit-BEGIN$
        suite.addTestSuite(FirstTestCase.class);
        suite.addTestSuite(SecondTestCase.class);
        //$JUnit-END$
        return suite;
    }
}
```

Running TestSuite

- Select TestSuite class
- From the Run menu select Run
→ Run As → JUnit Test
- This will run the test cases in
your TestSuite class
- You will then get a report in the
JUnit Window



Interesting Point

- The JUnit classes TestCase and TestSuite both implement the JUnit Test interface
- Therefore, you can add JUnit TestSuites to other TestSuites

```
public static Test suite() {  
    TestSuite suite = new TestSuite("Test for testing");  
    //$JUnit-BEGIN$  
    suite.addTestSuite(FirstTestCase.class);  
    suite.addTestSuite(SecondTestCase.class);  
    suite.addTest(AllTests.suite());  
    //$JUnit-END$  
    return suite;  
}
```

Summary

- You have learned:
 - What is JUnit
 - How it is integrated into Eclipse
 - How to write Test Cases
 - How to write Test Suites
 - How do use JUnit in Eclipse

Labs!



Lab: Using JUnit