Using NUnit

The Open Source Test Tool

garth@ggilmour.com

Unit Testing in .NET

- NUnit is the standard .NET code testing tool
  - It provides a framework for writing test classes
  - Each test class contains one or more test methods
  - Each test method contains one or more assertions
  - If an assertion fails then its test method fails
- NUnit started as a .NET port of JUnit 3
  - But it used the .NET language feature known as attributes
    - This was copied in Java 5 but called ‘annotations’
    - So NUnit and JUnit 4 are practically identical

```
namespace NUnitDemos {
        [TestFixture]
        public class MathTest {
                [SetUp]
                public void Begin() {
                        math = new Math(30, 20);
                }
                [Test]
                public void addition() {
                        Assert.AreEqual(50, math.add(), “ + failed!”);
                }
                [Test]
                public void subtraction() {
                        Assert.AreEqual(10, math.subtract(), “ - failed!”);
                }
                [Test]
                public void multiplication() {
                        Assert.AreEqual(600, math.multiply(), “ * failed!”);
                }
                private Math math;
        }
}
```

NUnit Attributes

Attribute	Target	Meaning
TestFixture	Class	A class which is a unit test for another class
SetUp	Method	A method to be run before each test
TearDown	Method	A method to be run after each test
Test	Method	Marks a method as a test method (methods starting with ‘test’ are also run as tests)
ExpectedException	Test Method	Signifies that an exception should be thrown by the test, as identified by a type object
Ignore	Test Method	Signifies that a test should be run but should not count towards failures (displayed in yellow)
TestFixtureSetUp	Method	A method to be run before testing begins (only available since version 2.1)
TestFixtureTearDown	Method	A method to be run after testing is completed (only available since version 2.1)

Using Assertions In NUnit

- Assertions are static method of the ‘Assert’ class
  - Prior to V2.1 the ‘Assertion’ class was used instead
  - The methods of ‘Assert’ are more powerful and concise
- Assertions let you validate your code
  - By testing boolean expressions
  - By comparing basic types
    - Note that double comparisons require a delta
  - By comparing pairs of references
    - Using equality of state and/or equality of reference
  - By comparing arrays of references

Assertions In NUnit

Assert a boolean condition
IsTrue(bool condition, string msg, params object[] args) IsTrue(bool condition, string msg) IsTrue(bool condition)
IsFalse(bool condition, string msg, params object[] args) IsFalse(bool condition, string msg) IsFalse(bool condition)
Assert that an object should (or should not) be null
IsNotNull(Object obj, string msg, params object[] args) IsNotNull(Object obj, string msg) IsNotNull(Object obj)
IsNull(Object obj, string msg, params object[] args) IsNull(Object obj, string msg) IsNull(Object obj)

Assertions In NUnit

Test if two references refer to the same object
AreSame(Object obj1, Object obj2, string msg, params object[] args)
AreSame(Object obj1, Object obj2, string msg)
AreSame(Object obj1, Object obj2)
Automatically fail a test
Fail(string msg, params object[] args)
Fail(string msg)
Fail()
Automatically ignore a test
Ignore(string msg, object[] args)
Ignore(string msg)
Ignore()

Assertions In NUnit

Test if two integers are equal

AreEqual(int no1, int no2, string msg, params object[] args)
AreEqual(int no1, int no2, string msg)
AreEqual(int no1, int no2)

Test if two objects are equal

AreEqual(Object obj1, Object obj2, string msg, params object[] args)
AreEqual(Object obj1, Object obj2, string msg)
AreEqual(Object obj1, Object obj2)

Test if floating point numbers are equal
AreEqual(float no1, float no2, float delta, string msg, params object[] args) AreEqual(float no1, float no2, float delta, string msg) AreEqual(float no1, float no2, float delta)
AreEqual(double no1, double no2, double delta, string msg, params object[] args) AreEqual(double no1, double no2, double delta, string msg) AreEqual(double no1, double no2, double delta)

Assertions In NUnit

Test if two decimals are equal

AreEqual(decimal no1, decimal no2, string msg, params object[] args)
AreEqual(decimal no1, decimal no2, string msg)
AreEqual(decimal no1, decimal no2)

Test if two arrays are equal

AreEqual(System.Array array1, System.Array array2, string msg, params object[] args)
AreEqual(System.Array array1, System.Array array2, string msg)
AreEqual(System.Array array1, System.Array array2)