

前言：

项目中进行Unit Test时，肯定会用到框架，因为这样能够更快捷、方便的进行测试。
.Net环境下的测试框架非常多，在这里只是对MS Test和NUnit Test进行一下比较，
因为这两个框架用的较多，也有大虾想过同时只是这两种框架的方法。

一、简介

MS Test框架是Visual Studio自带的测试框架，可以通过新建一个Unit Test Project工程，

也可以建一个Class Library，然后添加对Microsoft.VisualStudio.QualityTools.UnitTestFramework.dll的引用。

然后就是创建测试用例，进行测试即可。

NUnit Test框架是一个xUnit家族种的第4个主打产品，完全由C#语言来编写，支持所有的.Net语言。

使用NUnit框架，我们需要下载[安装包](#)，安装后使用独立客户端进行使用。使用方法与MS Test类似，

新建一个Class Library，然后添加对nunit.framework.dll的引用，最后创建测试用例，在客户端中进行测试。

二、特性标签对比

所有的测试框架都共享相同的核心特性：Test Declaration, Test Execution, and Assertions.

在.Net中一般使用特性标签来添加额外的信息，下面就是MS Test和NUnit在特性标签上不同的地方。

MS Test Attribute	NUnit Attribute	用途
[TestClass]	[TestFixture]	定义一个测试类，里面可以包含很多测试函数和初始化、销毁函数（以下所有标签和其他断言）。
[TestMethod]	[Test]	定义一个独立的测试函数。
[ClassInitialize]	[TestFixtureSetUp]	定义一个测试类初始化函数，每当运行测试类中的一个或多个测试函数时，这个函数将会在测试函数被调用前被调用一次（在第一个测试函数运行前会被调用）。
[ClassCleanup]	[TestFixtureTearDown]	定义一个测试类销毁函数，每当测试类中的选中的测试函数全部运行结束后运行（在最后一个测试函数运行结束后运行）。
[TestInitialize]	[SetUp]	定义测试函数初始化函数，每个测试函数运行前都会被调用一次。
[TestCleanup]	[TearDown]	定义测试函数销毁函数，每个测试函数执行完后都会被调用一次。

[AssemblyInitialize]	--	定义测试Assembly初始化函数，每当这个Assembly中的有测试函数被运行前，会被调用一次（在Assembly中第一个测试函数运行前会被调用）。
[AssemblyCleanup]	--	定义测试Assembly销毁函数，当Assembly中所有测试函数运行结束后，运行一次。（在Assembly中所有测试函数运行结束后被调用）
[DescriptionAttribute]	[Category]	定义标识分组。

三、同时支持 NUnit 和 MSTest

通过上面的介绍，我们可以看到这两个框架只是标签的写法不一样，它们的具体功能都是一样的。

所以可以通过宏判断来同时支持这两个框架。

1、添加两种框架的DLL

MSTest: Microsoft.VisualStudio.TestTools.UnitTesting.dll

NUnit : nunit.framework.dll

2、在测试前添加以下代码：



复制代码

```
#if !NUNIT
using Microsoft.VisualStudio.TestTools.UnitTesting;
using Category =
Microsoft.VisualStudio.TestTools.UnitTesting.DescriptionAttribute;
#else
using NUnit.Framework;
using TestClass = NUnit.Framework.TestFixtureAttribute;
using TestMethod = NUnit.Framework.TestAttribute;
using TestInitialize = NUnit.Framework.SetUpAttribute;
using TestCleanup = NUnit.Framework.TearDownAttribute;
using TestContext = System.Object;
using ClassCleanup = NUnit.Framework.TestFixtureTearDownAttribute;
using ClassInitialize = NUnit.Framework.TestFixtureSetUpAttribute;
#endif
```



复制代码

四、断言对比

同上，大部分都是一样的，有些地方不太一样而已。

MS Test Assert	NUnit Assert	用途
AreEqual		验证值相等
AreNotEqual		验证值不相等
AreSame		验证引用相等
AreNotSame		验证引用不相等
Inconclusive		暗示条件还未被验证
IsTrue		验证条件为真
IsFalse		验证条件为假
IsInstanceOfType	Assert.IsInstanceOf<>	验证实例匹配类型
IsNotInstanceOfType	Assert.IsNotInstanceOf<>	验证实例不匹配类型
IsNotNull		验证条件为NULL
IsNull		验证条件不为NULL
Fail		验证失败

另外还有一些是NUnit中的，但是MS Test框架中是没有的：

- Assert.IsNaN
- Assert.IsEmpty
- Assert.IsNotEmpty
- Assert.Greater
- Assert.GreaterOrEqual
- Assert.Less
- Assert.LessOrEqual
- Assert.IsAssignableFrom
- Assert.IsNotAssignableFrom
- Assert.Ignore
- CollectionAssert.IsEmpty
- CollectionAssert.IsNotEmpty
- StringAssert.AreEqualIgnoringCase
- StringAssert.IsMatch
- FileAssert.AreEqual
- FileAssert.AreNotEqual

五、其他

1、整合Visual Studio和NUnit

在Visual Studio 2010中，通过安装NUnit插件，可以不使用外部客户端，直接运行测试。有兴趣的可以看一下这篇帖子：[5分钟实现VS2010整合NUnit进行单元测试](#)。

当然，貌似在最新版本的VS2012中，安装过NUnit之后，就可以在不需要安装插件的情况下，

运行NUnit的测试。

2、运行时区别

看网上的帖子讲，NUnit不是并行执行测试的，所有的测试都是放在一个线程当中。

而MSTest中每个测试都被放在单独的线程当中。

3、关于ClassCleanup和TestFixtureTearDown

在NUnit中，TestFixtureTearDown在最后一个测试执行完毕后，马上执行。

而在MSTest中，ClassCleanup在AssemblyCleanup前执行，但是并不是最后一个测试完毕后马上执行。