عنوان: **بررسی میزان پوشش آزمون‌های واحد به کمک برنامه PartCover**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۱/۲۴ ۲۳:۴۵:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: Unit testing

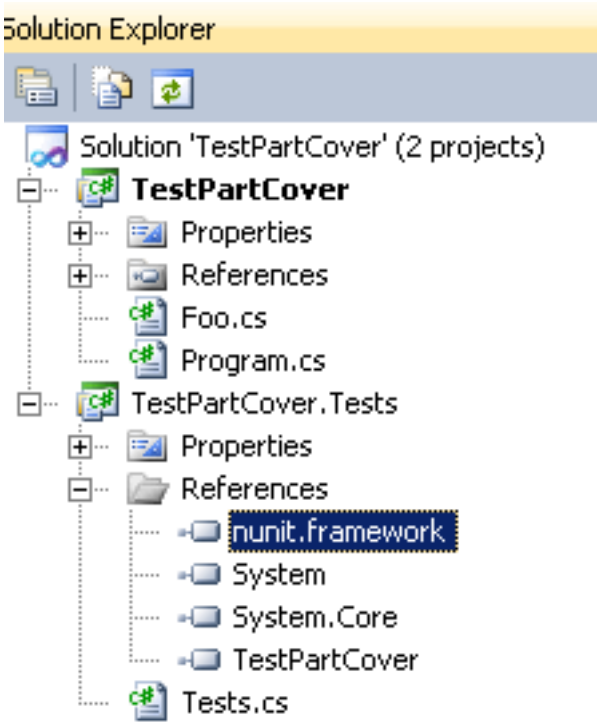
همیشه در حین توسعه‌ی یک برنامه این سؤالات وجود دارند:
- چند درصد از برنامه تست شده است؟
- برای چه تعدادی از متدهای موجود آزمون واحد نوشته‌ایم؟
- آیا همین آزمون‌های واحد نوشته شده و موجود، کامل هستند و تمام عملکردهای متدهای مرتبط را پوشش می‌دهند؟

این سؤالات به صورت خلاصه مفهوم Code coverage را در بحث Unit testing ارائه می‌دهند: برای چه قسمت‌هایی از برنامه آزمون واحد ننوشته‌ایم و میزان پوشش برنامه توسط آزمون‌های واحد موجود تا چه حدی است؟
بررسی این سؤالات در یک پروژه‌ی کم حجم، ساده بوده و به صورت بازبینی بصری ممکن است. اما در یک پروژه‌ی بزرگ نیاز به ابزار دارد. به همین منظور تعدادی برنامه جهت بررسی code coverage مختص پروژه‌های دات نتی تابحال تولید شده‌اند که در ادامه لیست آن‌ها را مشاهده می‌کنید:

NCover
Pex & Mole
DotCover
و ...

تمام این‌ها تجاری هستند. اما در این بین برنامه‌ی PartCover سورس باز و رایگان بوده و همچنین مختص به NUnit نیز تهیه شده است. این برنامه را از اینجا می‌توانید دریافت و نصب کنید. در ادامه نحوه‌ی تنظیم آن‌را بررسی خواهیم کرد:

الف) ایجاد یک پروژه آزمون واحد جدید
جهت توضیح بهتر سه سؤال مطرح شده در ابتدای این مطلب، بهتر است یک مثال ساده را در این زمینه مرور نمائیم: (پیشنیاز: (+))
یک Solution جدید در VS.NET آغاز شده و سپس دو پروژه جدید از نوع‌های کنسول و Class library به آن اضافه شده‌اند:



پروژه کنسول، برنامه اصلی است و در پروژه Class library ، آزمون‌های واحد برنامه را خواهیم نوشت.
کلاس اصلی برنامه کنسول به شرح زیر است:

```
namespace TestPartCover
{
public class Foo
{
    public int DoFoo(int x, int y)
    {
        int z = 0;
        if ((x > 0) && (y > 0))
        {
            z = x;
        }
        return z;
    }

    public int DoSum(int x)
    {
        return ++x;
    }
}
}
```

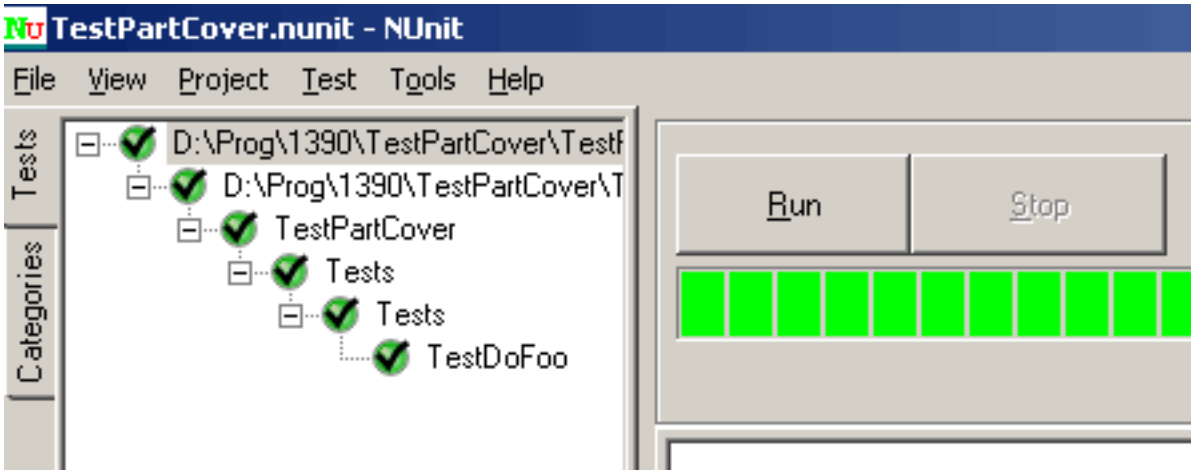
و کلاس آزمون واحد آن در پروژه class library مثلا به صورت زیر خواهد بود:

```
using NUnit.Framework;

namespace TestPartCover.Tests
{
[TestFixture]
public class Tests
{
    [Test]
    public void TestDoFoo()
    {
        var result = new Foo().DoFoo(-1, 2);
        Assert.That(result == 0);
    }
```

```
}
}
```

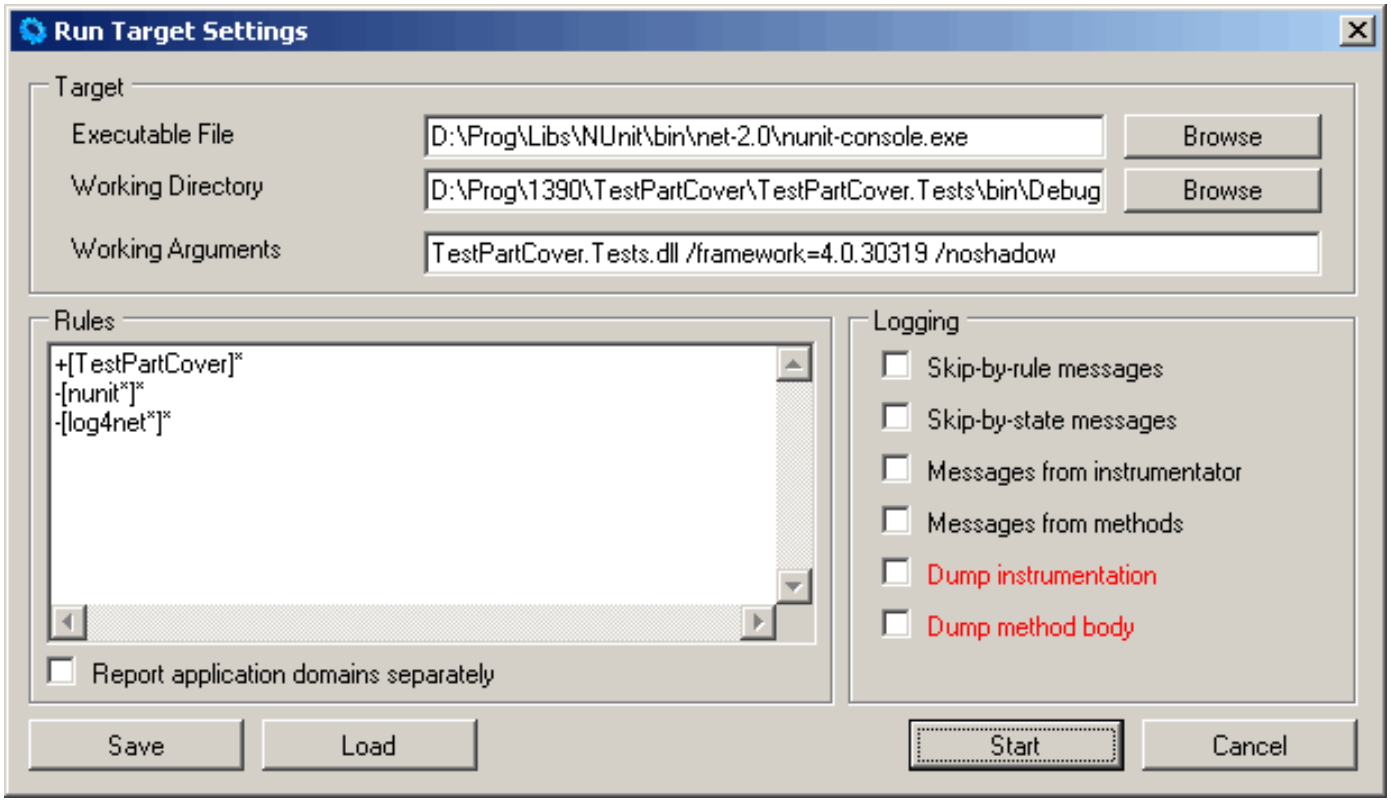
که نتیجه‌ی بررسی آن توسط NUnit test runner به شکل زیر خواهد بود:



به نظر همه چیز خوب است! اما آیا واقعا این آزمون کافی است؟!

ب) در ادامه به کمک برنامه‌ی PartCover می‌خواهیم بررسی کنیم میزان پوشش آزمون‌های واحد نوشته شده تا چه حدی است؟

پس از نصب برنامه، فایل PartCover.Browser.exe را اجرا کرده و سپس از منوی فایل، گزینه‌ی Run Target را انتخاب کنید تا صفحه‌ی زیر ظاهر شود:



توضیحات:

در قسمت executable file آدرس فایل nunit-console.exe را وارد کنید. این برنامه چون در حال حاضر برای دات نت 2 کامپایل شده امکان بارگذاری dll های دات نت 4 را ندارد. به همین منظور فایل nunit-console.exe.config را باز کرده و تنظیمات زیر را به آن اعمال کنید (مهم!):

```
<configuration>
<startup>
<supportedRuntime version="v4.0.30319" />
</startup>
```

و همچنین

```
<runtime>
<loadFromRemoteSources enabled="true" />
```

در ادامه مقابل working directory ، آدرس پوشه bin پروژه unit test را تنظیم کنید.

در این حالت working arguments به صورت زیر خواهند بود (در غیراینصورت باید مسیر کامل را وارد نمائید):

```
TestPartCover.Tests.dll /framework=4.0.30319 /noshadow
```

نام dll وارد شده همان فایل class library تولیدی است. آرگومان بعدی مشخص می‌کند که قصد داریم یک پروژه‌ی دات نت 4 را توسط NUnit بررسی کنیم (اگر ذکر نشود پیش فرض آن دات نت 2 خواهد بود و نمی‌تواند اسمبلی‌های دات نت 4 را بارگذاری کند). منظور از noshadow این است که NUnit مجاز به تولید shadow copies از اسمبلی‌های مورد آزمایش نیست. به این صورت

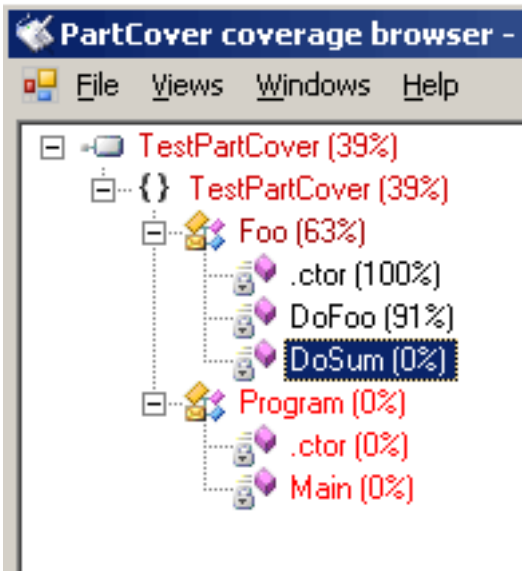
برنامه‌ی PartCover می‌تواند بر اساس StackTrace نهایی، سورس متناظر با قسمت‌های مختلف را نمایش دهد.
اکنون نوبت به تنظیم Rules آن است که یک سری RegEx هستند؛ به عبارتی چه اسمبلی‌هایی آزمایش شوند و کدام‌ها خیر:

```
+[TestPartCover]*
-[nunit*]*
-[log4net*]*
```

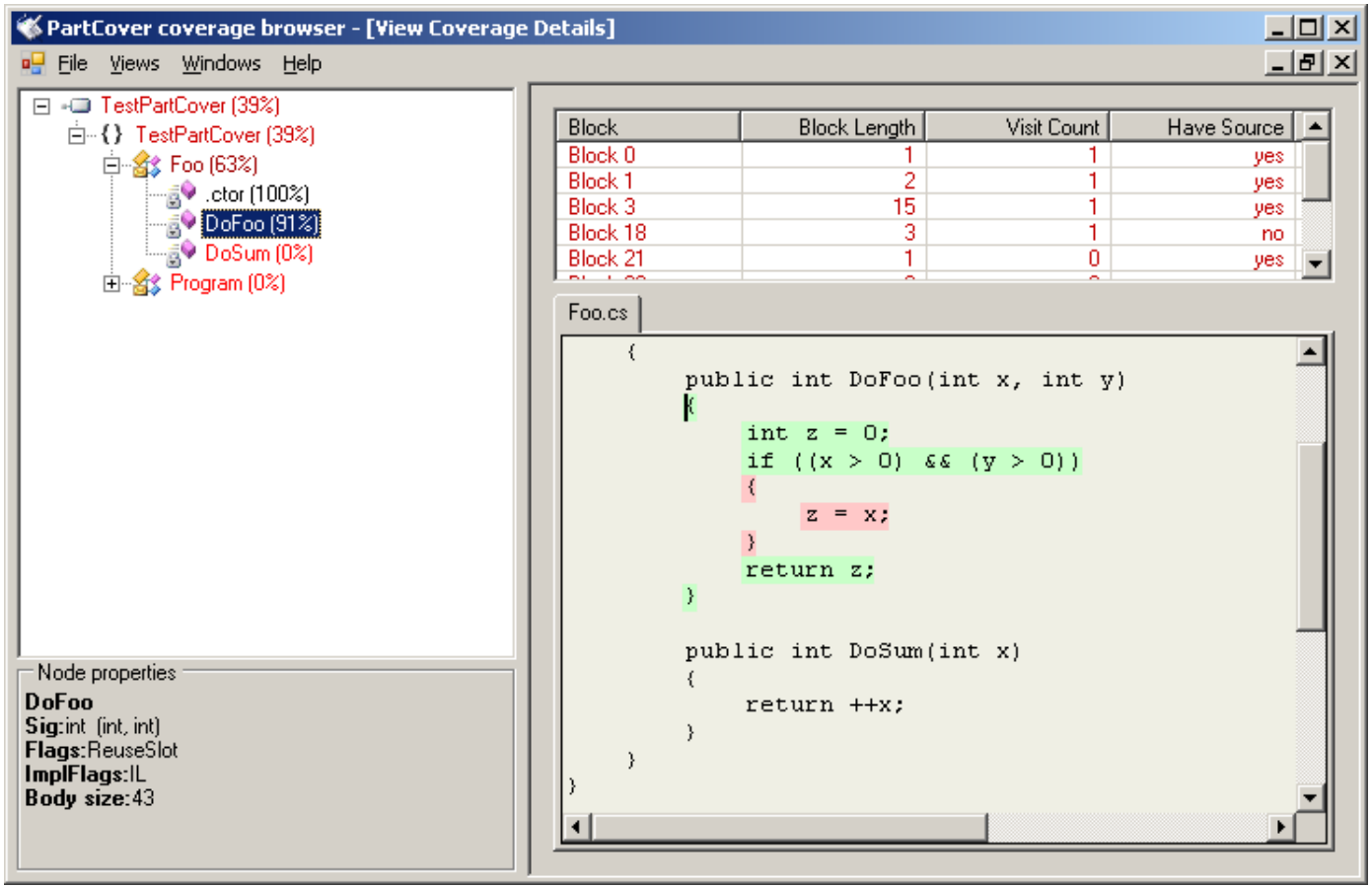
همانطور که ملاحظه می‌کنید در اینجا از اسمبلی‌های NUnit و log4net صرفنظر شده است و تنها اسمبلی TestPartCover (همان برنامه کنسول، نه اسمبلی برنامه آزمون واحد) بررسی خواهد گردید.
اکنون بر روی دکمه Save در این صفحه کلیک کرده و فایل نهایی را ذخیره کنید (بعدا توسط دکمه Load در همین صفحه قابل بارگذاری خواهد بود). حاصل باید به صورت زیر باشد:

```
<PartCoverSettings>
<Target>D:\Prog\Libs\NUnit\bin\net-2.0\nunit-console.exe</Target>
<TargetWorkDir>D:\Prog\1390\TestPartCover\TestPartCover.Tests\bin\Debug</TargetWorkDir>
<TargetArgs>TestPartCover.Tests.dll /framework=4.0.30319 /noshadow</TargetArgs>
<Rule>+[TestPartCover]*</Rule>
<Rule>-[nunit*]*</Rule>
<Rule>-[log4net*]*</Rule>
</PartCoverSettings>
```

برای شروع به بررسی، بر روی دکمه Start کلیک نمائید. پس از مدتی، نتیجه به صورت زیر خواهد بود:



بله! آزمون واحد تهیه شده تنها 39 درصد اسمبلی TestPartCover را پوشش داده است. مواردی که با صفر درصد مشخص شده‌اند، یعنی فاقد آزمون واحد هستند و نکته مهم‌تر پوشش 91 درصدی متد DoFoo است. برای اینکه علت را مشاهده کنید از منوی View ، گزینه‌ی Coverage detail را انتخاب کنید تا تصویر زیر نمایان شود:



قسمت نارنجی در اینجا به معنای عدم پوشش آن در متد TestDoFoo تهیه شده است. تنها قسمت‌های سبز را توانسته‌ایم پوشش دهیم و برای بررسی تمام شرط‌های این متد نیاز به آزمون‌های واحد بیشتری می‌باشد.

روش نهایی کار نیز به همین صورت است. ابتدا آزمون واحد تهیه می‌شود. سپس میزان پوشش آن بررسی شده و در ادامه سعی خواهیم کرد تا این درصد را افزایش دهیم.