

عنوان: استفاده یکپارچه از NUnit در VS.NET بدون نیاز به افزونه‌ها

نویسنده: وحید نصیری

تاریخ: ۱۳۹۰/۰۶/۱۷ ۱۱:۰۲:۰۰

آدرس: www.dotnettips.info

برچسب‌ها: Unit testing

برای استفاده ساده‌تر از ابزارهای [unit testing](#) در ویژوال استودیو [افزونه‌های](#) زیادی وجود دارند، از ری شارپر تا CodeRush تا حتی امکانات نسخه‌ی کامل VS.NET که با MSTest یکپارچه است. اما اگر نخواهیم از MSTest استفاده کنیم و همچنین افزونه‌ها را هم بخواهیم حذف کنیم (مثلا از نسخه‌ی رایگان express استفاده کنیم)، چطور؟

برای حل این مشکل چندین روش وجود دارد. یا می‌شود از test runner این‌ها استفاده کرد که اصلا نیازی به IDE ندارند و مستقل است؛ یا می‌توان به صورت زیر هم عمل کرد:

به خواص پروژه در VS.NET مراجعه کنید. برگه‌ی Build events را باز کنید. در اینجا می‌خواهیم post-build event را مقدار دهی کنیم. به این معنا که پس از هر build موفق، لطفا این دستورات خط فرمان را اجرا کن.

NUnit به همراه test runner خط فرمان هم ارائه می‌شود و نام آن nunit-console.exe است. اگر به محل نصب آن مراجعه کنید، عموماً در آدرس C:\Program Files\NUnit xyz\bin\nunit-console.exe قرار دارد. برای استفاده از آن تنها کافی است تنظیم زیر صورت گیرد:

```
c:\path\nunit-console.exe /nologo $(TargetPath)
```

TargetPath به صورت خودکار با نام اسمبلی جاری پروژه در زمان اجرا جایگزین می‌شود.

اکنون پس از هر Build، به صورت خودکار nunit-console.exe اجرا شده، اسمبلی برنامه که حاوی آزمون‌های واحد است به آن ارسال گردیده و سپس خروجی کار در output window نمایش داده می‌شود. اگر خطایی هم رخ داده باشد در قسمت errors قابل مشاهده خواهد بود.

در اینجا حتی بجای برنامه کنسول یاده شده می‌توان از برنامه nunit.exe هم استفاده کرد. در این حالت GUI اصلی پس از هر Build نمایش داده می‌شود:

```
c:\path\nunit.exe $(TargetPath)
```

چند نکته:

1- برنامه nunit-console.exe 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:

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

و همچنین:

```
<runtime>
```

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

2- خروجی نتایج اجرای آزمون‌ها را به صورت XML [هم می‌توان](#) ذخیره کرد. مثلاً:

```
c:\path\nunit-console.exe /xml:$(ProjectName)-tests.xml /nologo $(TargetPath)
```

3- از فایل xml ذکر شده می‌توان گزارشات زیبایی تهیه کرد. برای مثال:

[Generating Report for NUnit](#)

[NUnit2Report Task](#)

جهت مطالعه بیشتر:

[Setting up Visual C#2010 Express with NUnit](#)

[Use Visual Studio's Post-Build Events to Automate Unit Testing Running](#)

[Ways to Run NUnit From Visual Studio 3](#)

نظرات خوانندگان

نویسنده: علی

تاریخ: ۱۰:۵۵ ۱۳۹۲/۰۴/۱۰

با عرض سلام و تشکر از وب سایت پر محتواتون.

اونچه رو که در بالا فرموده بودین رو بنده آزمایش کردم ولی متاسفانه به هنگام build کردن پروژه با خطای زیر روبرو می‌شوم :

```
Error 1 The command "c:\program files\nunit 2.6.2\bin\nunit.exe C:\Users\Ali\documents\visual studio 2010\Projects\NUnitExample\NUnitExample.UnitTests\bin\Debug\NUnitExample.UnitTests.dll /run" exited with code 9009. NUnitExample.UnitTests
```

ممنون می‌شوم راهنماییم کنید.

نویسنده: وحید نصیری

تاریخ: ۱۱:۳۷ ۱۳۹۲/۰۴/۱۰

- علت اینجا است که در مسیر فایل dll شما space وجود دارد؛ در قسمت visual studio 2010 به همین منظور نیاز است برای اجرای یک دستور خط فرمان، این نوع مسیرها داخل "" قرار گیرند (یک اصل کلی است در مورد تمام فرامین خط فرمان).
- ضمناً اگر از VS 2012 استفاده می‌کنید، بهتر است از [NUnit Test Adapter](#) کمک بگیرید، تا با یک سیستم یکپارچه بتوانید کار کنید.