

عنوان: تولید خودکار آزمون‌های واحد NUnit

نویسنده: وحید نصیری

تاریخ: ۱۳۸۹/۱۱/۱۲ ۰۹:۴۳:۰۰

آدرس: www.dotnettips.info

برچسب‌ها: Unit testing

تعدادی ابزار برای تولید خودکار متدهای [آزمون‌های واحد](#) NUnit از روی کلاس‌های موجود در یک اسمبلی وجود دارند که به دو دسته تقسیم می‌شود:

الف) آن‌هایی که فقط نام کلاس‌های آزمون واحد و نام متدهای آن‌را به صورت خودکار تولید می‌کنند

[NStub](#)

[Top Coder .Net Test Generator](#)

[CodeSmith NUnit Test Generator](#)

این ابزارها و کتابخانه‌ها، تنها کاری که انجام می‌دهند یافتن کلاس‌ها و متدهای عمومی موجود در یک اسمبلی توسط Reflection و سپس تولید یک سری فایل آماده از روی این اطلاعات است. برای مثال اگر نام کلاس شما Class1 است فایلی به نام TestClass1 را تولید می‌کنند و اگر یکی از متدهای عمومی این کلاس به نام Method1 باشد، یک متد خالی را به نام Method1Test ایجاد خواهند کرد و همین.

تبدیل CodeSmith NUnit Test Generator فوق به یک [T4 template](#) کار ساده‌ای است.

ب) ابزارهایی که علاوه بر مورد الف، سعی می‌کنند بدنه‌ای را نیز برای متدهای واحد تولید شده تهیه کنند

[Novell NUnitGen AddIn](#)

[Edwinyeah TestGen.Net](#)

[NUnit Test Case Code Generator](#)

این افزونه‌ها و برنامه‌ها سعی می‌کنند به کمک Reflection و همچنین امکانات تولید کد موجود در VS.NET نسبت به تولید کلاس‌ها، متدها و بدنه‌های نمونه آن‌ها اقدام کنند. برای مثال اگر نام متد کلاسی، Method1 به همراه یک پارامتر از نوع int باشد، بدنه تولید شده به همراه وهله سازی از کلاس آن و فراخوانی این متد به همراه پارامتر آن خواهد بود. مشکل مهم این پروژه‌های سورس باز کوچک هم عدم تعهد به نگهداری آن‌ها است. برای مثال آخرین به روز رسانی موجود افزونه‌ی NUnitGen شرکت ناول، مخصوص VS2008 است یا آخرین به روز رسانی TestGen.Net مربوط به دات نت یک است (سورسی هم که در سایت سورس فورج قرار داده ناقص است) یا مقاله‌ی سایت CodeProject که ذکر گردید، با نگارش‌های جدید NUnit درست کار نمی‌کند و کامپایل نمی‌شود.

در بین این‌ها به نظر من Edwinyeah TestGen.Net کار جالبی را انجام داده است و چندین زبان را هم پشتیبانی می‌کند. البته همانطور که عنوان شد توانایی بارگذاری اسمبلی‌های نگارش‌های جدید دات نت را ندارد که موضوع مهمی نیست. سورس آن‌را می‌توان دریافت و سپس جهت دات نت 4 کامپایل کرد. البته یک سری از کلاس‌های آن هم که در سورس موجود نیستند را می‌شود از اسمبلی کامپایل شده‌ی آن با Reflector درآورد، به پروژه اصلی اضافه و سپس کامپایل کرد! کامپایل شده‌ی آن‌را جهت دات نت 4 [از اینجا](#) دریافت کنید.