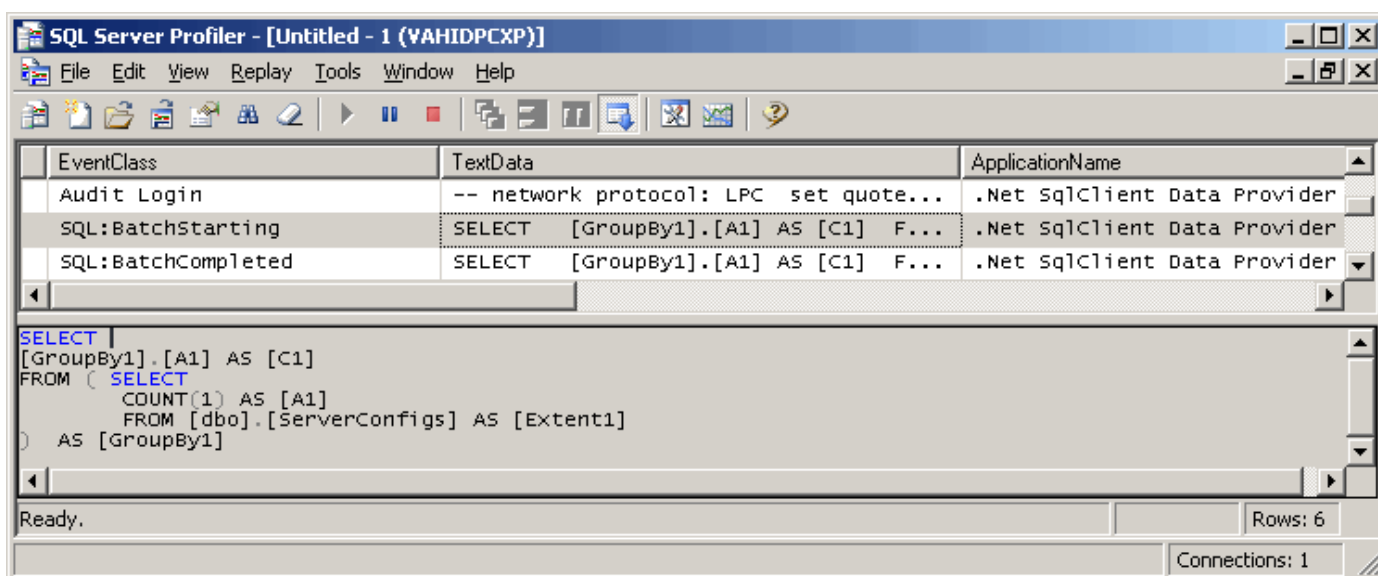


این روزها با وجود ORMs، کوئری SQL نوشتن شبیه به دورانی شده که با وجود زبان‌های سطح بالا، عده‌ای علاقمند هستند با استفاده از زبان اسمبلی برنامه نویسی کنند! WCF RIA Services به صورت پیش فرض از entity framework استفاده می‌کند (هر چند می‌توان از سایر ORMs هم استفاده کرد)، بنابراین عنوان صحیح‌تر بحث این خواهد بود: چگونه خروجی SQL تولید شده توسط Entity framework را بررسی کنیم؟

الف) استفاده از SQL Server profiler

اولین برنامه‌ای که از سال‌ها قبل، حتی پیش از ظهور ORMs وجود داشته، برنامه‌ی SQL server profiler است، که عموماً در مسیر ذیل قابل دستیابی است:

Start Menu->Programs->Microsoft SQL Server 2008->Performance Tools->SQL Server profiler

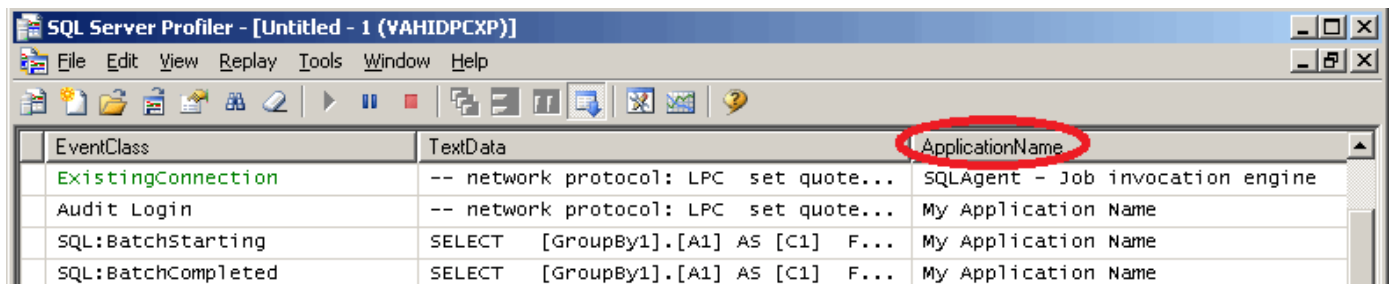


نکته مهم:

حین کار با SQL Server profiler، ممکن است انبوهی از کوئری‌های دیگر مثلاً مرتبط با SQL Server agent یا reporting services و غیره نیز لاگ شوند. اما الان ما تنها به کوئری‌های برنامه‌ی خود نیاز داریم. برای این منظور به کانکشن استرینگ خود، گزینه‌ی Application Name=My Application Name را نیز اضافه کنید:

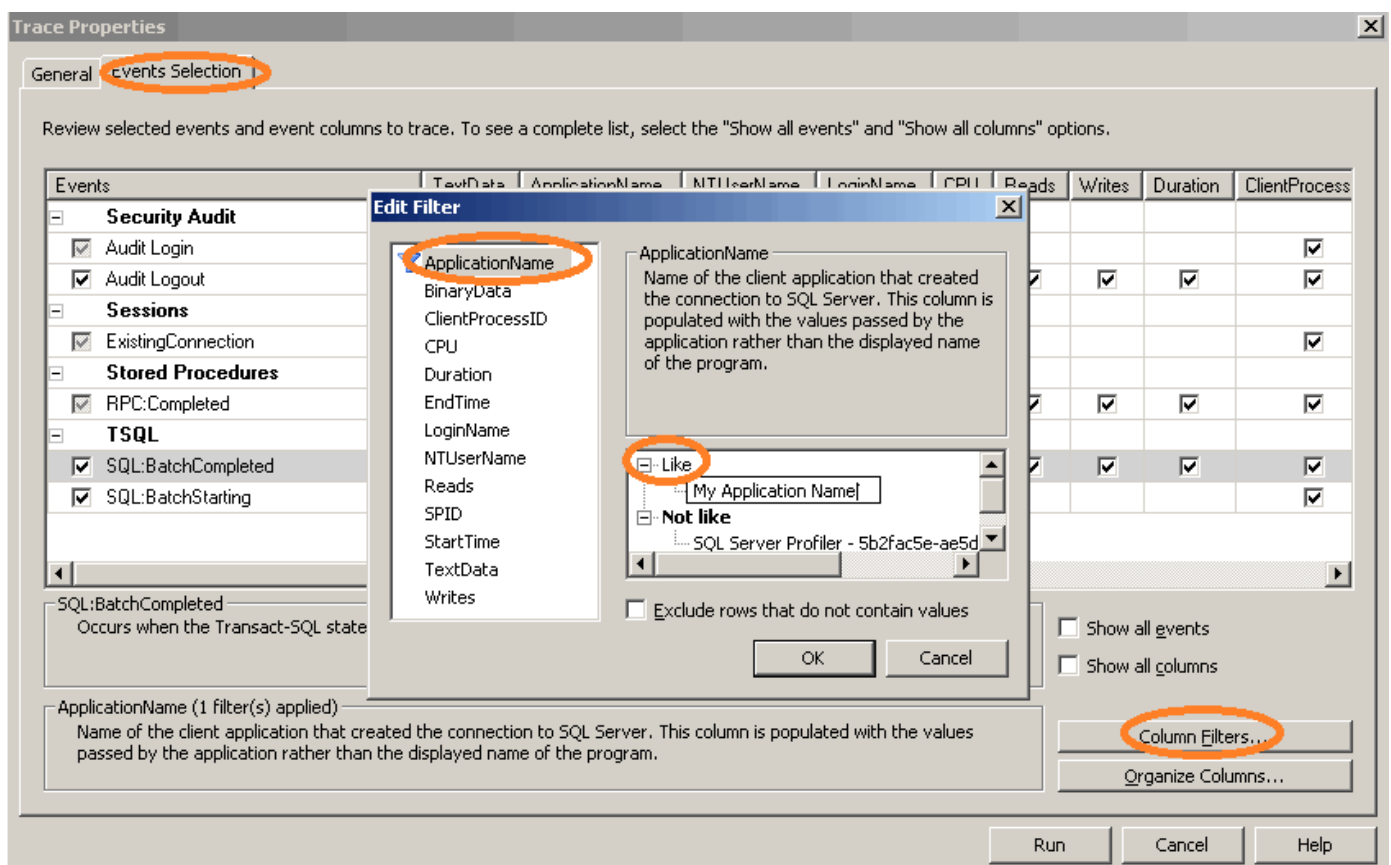
```
<connectionStrings>
  <add name="dmEntities"
  connectionString="metadata=res://*/Models.dmDataModel.csdl|res://*/Models.dmDataModel.ssdl|res://*/Models.dmDataModel.msl;provider=System.Data.SqlClient;provider connection string="Data Source=(local);Initial Catalog=dm;Integrated Security=True;Application Name=My Application Name;MultipleActiveResultSets=True" providerName="System.Data.EntityClient" />
</connectionStrings>
```

اکنون اگر برنامه را با پروفایلر مورد بررسی قرار دهید خروجی به صورت زیر خواهد بود:



EventClass	TextData	ApplicationName
ExistingConnection	-- network protocol: LPC set quote...	SQLAgent - Job invocation engine
Audit Login	-- network protocol: LPC set quote...	My Application Name
SQL:BatchStarting	SELECT [GroupBy1].[A1] AS [C1] F...	My Application Name
SQL:BatchCompleted	SELECT [GroupBy1].[A1] AS [C1] F...	My Application Name

برای فیلتر کردن Application Name مورد نظر، در ابتدای کار که یک سشن جدید را آغاز می‌کنید به برگه‌ی events selection مراجعه کرده و بر روی دکمه‌ی column filter کلیک کنید. گزینه‌ی application name را در صفحه‌ی باز شده انتخاب نموده و در قسمت Like آن مطابق تصویر زیر، نام برنامه‌ی خود را وارد نمایید:



Trace Properties - [Untitled - 1 (VAHIDPCXP)]

General **Events Selection**

Review selected events and event columns to trace. To see a complete list, select the "Show all events" and "Show all columns" options.

Events:

- ☒ Security Audit
  - ☒ Audit Login
  - ☒ Audit Logout
- ☒ Sessions
  - ☒ ExistingConnection
- ☒ Stored Procedures
  - ☒ RPC:Completed
- ☒ TSQL
  - ☒ SQL:BatchCompleted
  - ☒ SQL:BatchStarting

ApplicationName (1 filter(s) applied)  
Name of the client application that created the connection to SQL Server. This column is populated with the values passed by the application rather than the displayed name of the program.

**Edit Filter**

ApplicationName

Name of the client application that created the connection to SQL Server. This column is populated with the values passed by the application rather than the displayed name of the program.

☒ Like  
My Application Name

☐ Not like  
SQL Server Profiler - 5b2fac5e-ae5d

☐ Exclude rows that do not contain values

OK Cancel

☐ Show all events  
☐ Show all columns

**Column Filters...**

Organize Columns...

Run Cancel Help

ب) استفاده از IntelliTrace در VS.NET 2010

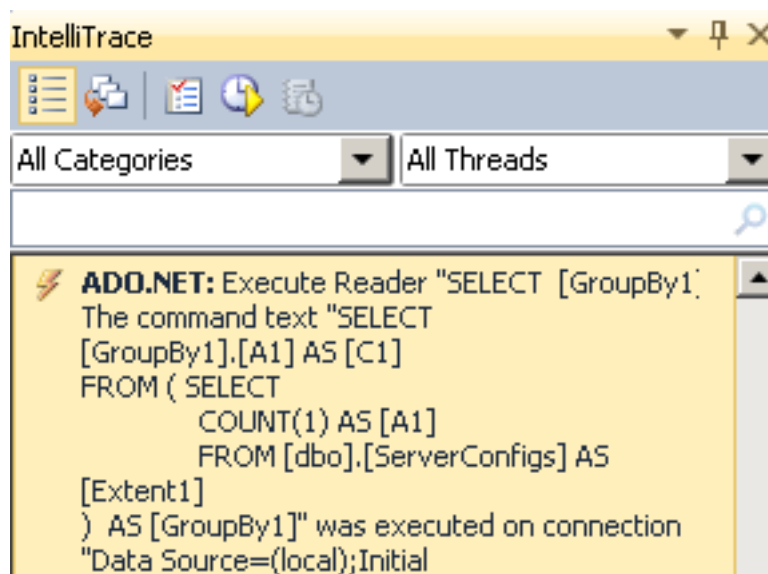
برنامه را در حالت دیباگ در VS.NET 2010 اجرا کنید. در هر لحظه‌ای می‌توان روی گزینه‌ی Break all کلیک کرد و خروجی SQL

تولید شده را نیز علاوه بر اطلاعات دیگر مشاهده نمود:



To view IntelliTrace data, you must break execution of your application.

 **Break All**



ج) استفاده از برنامه‌ی حرفه‌ای [entity framework profiler](#) این برنامه از هر دو مورد قبل کاملتر بوده و اساساً برای لاگ کردن کوئری‌ها، مدت زمان اجرا، گزارشگیری از وضعیت برنامه، کدامیک از کوئری‌ها سنگین‌تر هستند، حتی از طریق کدام متد فراخوانی شده‌اند، ارائه‌ی گزارشات و راهنمایی‌هایی در مورد چگونگی بهبود کارایی برنامه‌ی تهیه شده و امثال آن کاربرد دارد.

استفاده از آن هم بسیار ساده است. ابتدا ارجاعی را به اسمبلی `HibernatingRhinos.Profiler.Appender.v4.0` به پروژه‌ی ASP.NET خود اضافه کنید (همان پروژه‌ی هوست مربوط به WCF RIA Service ما). سپس به فایل `Global.asax.cs` برنامه مراجعه کرده و یک سطر ذیل را اضافه کنید:

```
protected void Application_Start(object sender, EventArgs e)
{
    HibernatingRhinos.Profiler.Appender.EntityFramework.EntityFrameworkProfiler.Initialize();
}
```

از این پس تنها کافی است برنامه‌ی پروفایلر در حال اجرا بوده و برنامه شما نیز اجرا شود. کلیه‌ی تبدلات با دیتابیس لاگ خواهند شد.

## نظرات خوانندگان

نویسنده: محمود رمضانی  
تاریخ: ۱۳۸۹/۰۵/۲۹ ۱۳:۳۹:۳۸

سلام. یک سوال در زمینه EF ؟  
اینکه من همه کد ها رو به صورت Stored Procedure بنویسم و بعد فقط اون ها رو در EF استفاده کنم اشتباه؟  
از این بابت میگم اشتباه چون فکر می کنم با این کار از قدرت EF استفاده نمی کنم.

نویسنده: وحید نصیری  
تاریخ: ۱۳۸۹/۰۵/۲۹ ۱۴:۳۴:۴۴

سلام،  
من یک زمانی از طرفداران نوشتن stored procedure بودم اما الان به دلایل ذیل نیستم:  
- امنیت: تمام کوئری‌های تولیدی entity framework از نوع پارامتری هستند. این مورد یعنی عقیم سازی حملات تزریق اس کیوال . (بنابراین اگر کسی عنوان کند که با SP ما امنیت بیشتری داریم، باید عنوان کرد که اینجا هم به همین صورت است)  
- سرعت: در نگارش‌های جدید اس کیوال سرور، با کوئری‌های پارامتری دقیقاً مانند SP رفتار می‌شود. همان کش شدن execution plan و غیره. بنابراین اینجا هم از همان مزایای SP برخوردار هستیم و سرعت سیستم مطلوب است.  
- مشکلات نگهداری SP :  
شما می‌تونید ساختار جداول رو تغییر بدید بدون اینکه اس کیوال سرور به شما پیغام خطایی در مورد غیر معتبر شدن یک SP بدهد. شما می‌تونید حتی یک SP غیر معتبر را تا زمانیکه syntax مربوط به آن صحیح است تولید کنید. هر دو مورد در زمان اجرا، سبب از کار افتادن برنامه می‌شوند. اما با EF این مشکلات را ندارید. ساختار را که عوض کنید برنامه دیگر کامپایل نمی‌شود. این مورد در برنامه‌های بزرگ خیلی خیلی خوب است!  
مورد دیگر: یک برنامه بزرگ با چند صد SP رو در نظر بگیرید. جدا نگهداری این‌ها پیدا کردن کدها در یک برنامه‌ی بزرگ عذاب است. طبقه بندی آن‌ها یک طرف، اعمال تغییرات از طرف دیگر.  
- مورد دیگر که من دیدم در یک سری از سایت‌ها در مورد آن بحث می‌کنند این است که نباید business logic برنامه را داخل دیتابیس طراحی کرد. این مورد باید با کد نویسی داخل برنامه باشد. در اینجا هم باز EF یا موارد مشابه بهتر هستند.  
- مورد دیگری که در SP ها مشکل ساز می‌شود به اشتراک گذاری آن در بین برنامه‌های مختلف است. هم خوب است. بالاخره کد نویسی کمتر می‌شود. هم بد است، از این لحاظ که شاید این SP نیاز به تغییر داشت و اینجا است که برنامه‌های دیگر مشکل پیدا می‌کنند.

نویسنده: وحید نصیری  
تاریخ: ۱۳۸۹/۰۵/۲۹ ۱۴:۳۸:۴۷

- یک مورد دیگر را هم اضافه کنم. با SP نویسی شما برنامه خودتون رو به SQL Server گره می‌زنید. اما با استفاده از EF خالص به سادگی می‌تونید به بانک‌های اطلاعاتی دیگری که پروایدر EF دارند سوئیچ کنید.

نویسنده: محمود رمضانی  
تاریخ: ۱۳۸۹/۰۵/۳۰ ۲۰:۳۴:۵۸

ممنون بخاطر جواب.  
یک سوال دیگه اینکه EF موجب افزایش Round Trip نمی شه؟ منظورم در مقایسه با SP هست.  
البته دلایلی که ذکر کردید به شدت وسوسه کننده بود. این رو فقط برای دونستن پرسیدم.

نویسنده: وحید نصیری  
تاریخ: ۱۳۸۹/۰۵/۳۰ ۲۲:۵۰:۵۴

این مورد از چند دیدگاه قابل بررسی است:  
- بحث lazy loading و eager loading : بارگذاری کردن یا نکردن اطلاعات وابسته به یک شیء تا زمانیکه به آن‌ها نیاز است یا

اینکه هر آنچه نیاز است به یکباره load شود. با eager loading می‌شود تعداد رفت و برگشت‌ها به دیتابیس را در شرایطی که به آن نیاز است، به حداقل رساند.

- بحث دیگر Deferred Execution است (که با lazy loading متفاوت است) و اساس LINQ را تشکیل می‌دهد، فارغ از اینکه LINQ to SQL است یا LINQ to Entities یا ... . زمانیکه شما یک کوئری LINQ را می‌نویسید این کوئری بر روی دیتابیس اجرا نمی‌شود. این اجرا تا زمانیکه شما به اطلاعات آن نیازی نداشته باشید به تأخیر خواهد افتاد. به عبارتی اگر شما 20 عبارت LINQ را پشت سر هم تشکیل دهید حتی یک رفت و برگشت هم به دیتابیس نخواهید داشت. فقط expression tree آن تشکیل می‌شود. همینجا است که مفهوم وجودی SubmitChanges بهتر مشخص می‌شود. شما 10 عبارت تعریف رکوردهای جدید را با LINQ تشکیل دهید. هیچ اتفاقی بر روی دیتابیس رخ نخواهد داد (Deferred Execution). زمانیکه SubmitChanges فراخوانی می‌شود این expression tree نهایی به کدهای SQL متناظر ترجمه خواهد شد.

اطلاعات بیشتر : [+](#)

نویسنده: Meysam  
تاریخ: ۱۳۸۹/۰۵/۳۱ ۰۰:۳۶:۰۱

- امنیت: دلیل امنیت فقط در تشخیص SQL-Injection نیست! اگر با ابزارهایی مثل acunetix سایت رو تست کنید، میبینید که همیشه یک threshold در حد MS درآورد که همیشه تشخیص اینجکشن رو داد. بله EF و SP این اشتراک رو دارن که همیشه اینجکت کرد ولی همیشه Vulnerability رو تشخیص داد.

- شخصا به خاطر مشکلات Deployment سعی میکنم BL بیشتر در SP باشه.

- در مورد تغییرات هم عرض کنم، ابزارهایی مثل SQL-ToolBelt مشکل رو حل کردن.

- اینکه برنامه به tightly couple نشه خوبه.

- Round Trip زیاده! به طور مثال فرض کنید یک مقدار تولیدی برای یک فیلد که کلید اصلی است (GUID یا Identity) البته فنون شرقی دارن ولی راه حل استاندارد نیستن!

- تمامی مباحث Loading درست ولی باید یک Trade Off بین بزرگی DB، سرعت نگاشت، قابل اطمینان بودن! (EF برخی قابلیت ها رو نداره اینجا رو نگاه کنید <http://blogs.msdn.com/b/alexj/archive/2009/03/26/index-of-tips.aspx> البته انتظار که ندارید سازنده ش مستقیم بگه نمیشه!)

بحث Change Tracking واقعا عالیه ولی به طور کلی، در یک برنامه Enterprise همیشه از EF استفاده کرد (به همراه OData)

مخلصیم

نویسنده: وحید نصیری  
تاریخ: ۱۳۸۹/۰۵/۳۱ ۰۱:۴۰:۳۷

- اصلا راجع به تشخیص بحث نشد. نه SP و نه EF و نه L2SQL و نه NHibernate هیچکدام روش تشخیصی برای درک اینکه آیا ورودی فعلی خطرناک است یا خیر ندارند. کوئری پارامتری است که اینها را حفظ می‌کند: [+](#)

- در مورد توزیع از چه نوع؟ برنامه ویندوزی یا برنامه وب؟ برنامه‌های ویندوزی از نظر من اصلا برای یک سازمان مناسب نیستند : [+](#)

- مثال Round Trip اصلا واضح نبود ...

- مطلبی که لینک دادید بیشتر مربوط به نگارش اول EF بود و الان خیلی تغییر کرده و حالا اگر با EF مشکل دارید از NHibernate استفاده کنید یا موارد دیگر.

- مورد آخر بیشتر یک نظر شخصی است (و محترم) ولی دلایل آن قانع کننده نبودند.

نویسنده: Meysam  
تاریخ: ۱۳۸۹/۰۵/۳۱ ۱۳:۳۹:۰۰

همونطور که قبلا عرض کردم، فقط تشخیص اینجکشن نیست (یا به عبارتی اینکه نشه اینجکشن کرد). نکته اصلی اینجاست که مقدار زمان لازم در زمانی که اینجکشن شده تفاوت داره و در SP هیچ تفاوتی بین حالت عادی و حالتی که اینجکشن شده وجود نداره و این مهم از طریق برخی ابزارها قابل تشخیصه (ولو با زمان زیاد).

- بله توزیع ویندوزی مناسب نیست ولی خب گاهی اوقات اجبار از طرف کارگزار وجود داره!  
مثال:- <http://blogs.msdn.com/b/adonet/archive/2010/06/28/performance-impact-of-server-side-generated-guids-in-ef.aspx>  
-اینکه EF از تمامی مشابه های موجود بهتره شکی ندارم.

نویسنده: Msanaei  
تاریخ: ۱۳۹۰/۰۱/۲۲ ۰۸:۰۱:۰۱

سلام  
ممکنه بفرمایید نسخه مجانی پروفایلر EF هم وجود داره یا نه؟ اگه بله از کجا میشه دانلود کرد ممنون.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۰/۰۱/۲۲ ۰۸:۱۶:۲۸

مجانی نیست ولی می‌تونید از آدرس زیر یک مجوز رایگان یک ماهه دریافت کنید. به آدرس ایمیل شما ارسال خواهد شد  
<http://efprof.com/Download>