عنوان: **خلاصه‌ای در مورد SQL Server CE**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۰۵ ۱۰:۱۶:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: SQL Server CE

SQL Server CE
برای اولین بار جهت استفاده در SmartPhones طراحی شد؛ جزو خانواده‌ی Embedded databases قرار می‌گیرد و این مزایا را دارد:
- نیازی به نصب ندارد و از چند DLL تشکیل شده است (برای مثال جهت استفاده در کارهای تک کاربره‌ی قابل حمل ایده‌آل است).
- رایگان است (جهت استفاده در کارهای تجاری و غیرتجاری).
- حجم کمی دارد (جمعا کمتر از دو مگابایت).
- پرویدر ADO.NET آن موجود است (توسط فضای نام System.Data.SqlServerCe که به کمک اسمبلی System.Data.SqlServerCe.dll قرار گرفته در مسیر C:\Program Files\Microsoft SQL Server Compact Edition\v3.5\Desktop ارائه می‌شود).
- با کمک ORM هایی مانند Entity framework و یا NHibernate نیز می‌توان با آن کار کرد.
- نسخه‌ی 4 نهایی آن که قرار است در زمان ارائه‌ی SP1 مربوط به VS.NET 2010 ارائه شود، جهت استفاده در برنامه‌های ASP.NET (برنامه‌های چند کاربره) ایی که تعداد کاربر کمی دارند، بهینه سازی شده و این مورد یک مزیت مهم نسبت به SQLite است که اساسا با تردهای همزمان جهت کار با بانک اطلاعاتی مشکل دارد.
- امکان گذاشتن کلمه‌ی عبور بر روی بانک اطلاعاتی آن وجود دارد که سبب رمزنگاری خودکار آن نیز خواهد شد (این مورد به صورت پیش فرض در SQLite پیش بینی نشده و جزو مواردی که است که باید برای آن هزینه کرد). الگوریتم رمزنگاری آن به صورت رسمی معرفی نشده، ولی به احتمال زیاد AES می‌باشد.
- از Sync Framework ADO.NET پشتیبانی می‌کند.

ملاحظات:
- به آن می‌توان به صورت نسخه‌ی تعدیل شده‌ی SQL Server 2000 با توانایی‌های کاهش یافته نگاه کرد. در آن خبری از رویه‌های ذخیره شده، View ها ، CLR Procs، CLR Triggers ، Full text search و غیره نیست (سطح توقع را باید در حد همان 2 مگابایت پایین نگه داشت!). لیست کامل : (+)
- Management studio مربوط به SQL Server 2005 به هیچ عنوان از آن پشتیبانی نمی‌کند و تنها نسخه‌ی 2008 است که نگارش 3 و نیم آن‌را پشتیبانی می‌کند آن هم نه با توانایی‌هایی که جهت کار با SQL Server اصلی وجود دارد. مثلا امکان rename یک فیلد را ندارد و باید برای اینکار کوئری نوشت . خوشبختانه یک سری پروژه‌ی رایگان در سایت CodePlex این نقایص را پوشش داده‌اند؛ برای مثال : ExportSqlCe
- از آنجائیکه DLL های SQL CE از نوع Native هستند، باید دقت داشت که حین استفاده از آن‌ها در دات نت فریم ورک اگر platform target قسمت build برنامه بر روی ALL CPU تنظیم شده باشد، برنامه به احتمال زیاد در سیستم‌های 64 بیتی کرش خواهد کرد (اگر در حین توسعه برنامه از DLLهای بومی 32 بیتی آن استفاده شده باشد). بنابراین نیاز است DLL های 64 بیتی را به صورت جداگانه جهت سیستم‌های 64 بیتی ارائه داد. اطلاعات بیشتر: (+) و (+) و (+)
- Entity framework یک سری از قابلیت‌های این بانک اطلاعاتی را پشتیبانی نمی‌کند. برای مثال اگر یک primary key از نوع identity را تعریف کردید، برنامه کار نخواهد کرد! لیست مواردی را که پشتیبانی نمی‌شوند، در این آدرس می‌توان مشاهده کرد.

و اخبار مرتبط با SQL CE را در این بلاگ می‌توانید دنبال کنید.

عنوان: **استفاده از SQL-CE به کمک NHibernate**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۲/۲۵ ۲۳:۴۹:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: SQL Server CE

خلاصه‌ای را در مورد SQL Server CE قبلا در این سایت مطالعه کرده‌اید . در ادامه خلاصه‌ای کاربردی را از تنظیمات و نکات مرتبط به کار با SQL-CE به کمک NHibernate ملاحظه خواهید نمود:

1) دریافت SQL-CE 4.0

Microsoft SQL Server Compact 4.0

همین مقدار برای استفاده از SQL-CE 4.0 به کمک NHibernate کفایت می‌کند و حتی نیازی به نصب سرویس پک یک VS 2010 هم نیست.

2) ابزار سازی جهت ایجاد یک بانک اطلاعاتی خالی SQL-CE

```
using System;
using System.IO;

namespace NHibernate.Helper.DbSpecific
{
    public class SqlCEDbHelper
    {
        const string engineTypeName = "System.Data.SqlServerCe.SqlCeEngine, System.Data.SqlServerCe";

        /// <summary>
        /// note: this method will delete existing db and then creates a new one.
        /// </summary>
        /// <param name="filename"></param>
        /// <param name="password"></param>
        public static void CreateEmptyDatabaseFile(string filename, string password = "")
        {
            if (File.Exists(filename))
                File.Delete(filename);

            var type = System.Type.GetType(engineTypeName);
            var localConnectionString = type.GetProperty("LocalConnectionString");
            var createDatabase = type.GetMethod("CreateDatabase");

            var engine = Activator.CreateInstance(type);

            string connectionStr = string.Format("Data Source='{0}';Password={1};Encrypt
Database=True", filename, password);
            if (string.IsNullOrWhiteSpace(password))
                connectionStr = string.Format("Data Source='{0}'", filename);

            localConnectionString.SetValue(
                obj: engine,
                value: connectionStr,
                index: null);
            createDatabase.Invoke(engine, new object[0]);
        }

        /// <summary>
        /// use this method to compact or encrypt existing db or decrypt it to a new db with all
records
        /// </summary>
        /// <param name="sourceConnection"></param>
        /// <param name="destConnection"></param>
        public static void CompactDatabase(string sourceConnection, string destConnection)
        {
            var type = System.Type.GetType(engineTypeName);
            var engine = Activator.CreateInstance(type);
```

```
            var localConnectionString = type.GetProperty("LocalConnectionString");
            localConnectionString.SetValue(
                obj: engine,
                value: sourceConnection,
                index: null);

            var compactDatabase = type.GetMethod("Compact");
            compactDatabase.Invoke(engine, new object[] { destConnection });
        }
    }
}
```

کلاس فوق، یک کلاس عمومی است و مرتبط به NHibernate نیست و در همه جا قابل استفاده است.
متد CreateEmptyDatabaseFile یک فایل بانک اطلاعاتی خالی با فرمت مخصوص SQL-CE را برای شما تولید خواهد کرد. به این ترتیب می‌توان بدون نیاز به ابزار خاصی، سریعا یک بانک خالی را تولید و شروع به کار کرد. در این متد اگر کلمه عبوری را وارد نکنید، بانک اطلاعاتی رمزنگاری شده نخواهد بود و اگر کلمه عبور را وارد کنید، دیتابیس اولیه به همراه کلیه اعمال انجام شده بر روی آن در طول زمان، با کمک الگوریتم AES به صورت خودکار رمزنگاری خواهند شد. کل کاری را هم که باید انجام دهید ذکر این کلمه عبور در کانکشن استرینگ است.
متد CompactDatabase، یک متد چند منظوره است. اگر بانک اطلاعاتی SQL-CE رمزنگاری نشده‌ای دارید و می‌خواهید کل آن‌را به همراه تمام اطلاعات درون آن رمزنگاری کنید، می‌توانید جهت سهولت کار از این متد استفاده نمائید. آرگومان اول آن به کانکشن استرینگ بانکی موجود و آرگومان دوم به کانکشن استرینگ بانک جدیدی که تولید خواهد شد، اشاره می‌کند.
همچنین اگر یک بانک اطلاعاتی SQL-CE رمزنگاری شده دارید و می‌خواهید آن‌را به صورت یک بانک اطلاعاتی جدید به همراه تمام رکوردهای آن رمزگشایی کنید، باز هم می‌توان از این متد استفاده کرد. البته بدیهی است که کلمه عبور را باید داشته باشید و این کلمه عبور جایی درون فایل بانک اطلاعاتی ذخیره نمی‌شود. در این حالت در کانکشن استرینگ اول باید کلمه عبور ذکر شود و کانکشن استرینگ دوم نیازی به کلمه عبور نخواهد داشت.

فرمت کلی کانکشن استرینگ SQL-CE هم به شکل زیر است:

```
Data Source=c:\path\db.sdf;Password=1234;Encrypt Database=True
```

البته این برای حالتی است که قصد داشته باشید بانک اطلاعاتی مورد استفاده را رمزنگاری کنید یا از یک بانک اطلاعاتی رمزنگاری شده استفاده نمائید. اگر بانک اطلاعاتی شما کلمه عبوری ندارد، ذکر Data Source=c:\path\db.sdf کفایت می‌کند.

این کلاس هم از این جهت مطرح شد که NHibernate می‌تواند ساختار بانک اطلاعاتی را بر اساس تعاریف نگاشت‌ها به صورت خودکار تولید و اعمال کند، «اما» بر روی یک بانک اطلاعاتی خالی SQL-CE از قبل تهیه شده (در غیراینصورت خطای The database file cannot be found. Check the path to the database را دریافت خواهید کرد).

نکته:
اگر دقت کرده باشید در این کلاس engineTypeName به صورت رشته ذکر شده است. چرا؟
علت این است که با ذکر engineTypeName به صورت رشته، می‌توان از این کلاس در یک کتابخانه عمومی هم استفاده کرد، بدون اینکه مصرف کننده نیازی داشته باشد تا ارجاع مستقیمی را به اسمبلی SQL-CE به برنامه خود اضافه کند. اگر این ارجاع وجود داشت، متدهای یاد شده کار می‌کنند، در غیراینصورت در گوشه‌ای ساکت و بدون دردسر و بدون نیاز به اسمبلی خاصی برای روز مبادا قرار خواهند گرفت.

3) ابزار مرور اطلاعات بانک اطلاعاتی SQL-CE

با استفاده از management studio خود SQL Server هم می‌شود با بانک‌های اطلاعاتی SQL-CE کار کرد، اما ... اینبار برخلاف نگارش کامل اس کیوال سرور، با یک نسخه‌ی بسیار بدوی، که حتی امکان rename فیلدها را هم ندارد مواجه خواهید شد. به همین جهت به شخصه برنامه SqlCe40Toolbox را ترجیح می‌دهم و اطمینان داشته باشید که امکانات آن برای کار با SQL-CE از امکانات ارائه شده توسط management studio مایکروسافت، بیشتر و پیشرفته‌تر است!

4) تنظیمات NHibernate جهت کار با SQL-CE

الف) پس از نصب SQL-CE ، فایل‌های آن‌را در مسیر C:\Program Files\Microsoft SQL Server Compact Edition\v4.0 می‌توان یافت. درایور ADO.NET آن هم در مسیر C:\Program Files\Microsoft SQL Server Compact Edition\v4.0\Desktop قرار دارد. بنابراین در ابتدا نیاز است تا ارجاعی را به اسمبلی System.Data.SqlServerCe.dll به برنامه خود اضافه کنید (نام پوشه desktop آن هم غلط انداز است. از این جهت که نگارش 4 آن، به راحتی در برنامه‌های ذاتا چند ریسمانی ASP.Net بدون مشکل قابل استفاده است).
نکته مهم: در این حالت NHibernate قادر به یافتن فایل درایور یاد شده نخواهد بود و پیغام خطای «Could not create the driver from NHibernate.Driver.SqlServerCeDriver» را دریافت خواهید کرد. برای رفع آن، اسمبلی System.Data.SqlServerCe.dll را در لیست ارجاعات برنامه یافته و در برگه خواص آن، خاصیت «Copy Local» را true کنید. به این معنا که NHibernate این اسمبلی را در کنار فایل اجرایی برنامه شما جستجو خواهد کرد.

ب) مطلب بعد، تنظیمات ابتدایی NHibernate است جهت شناساندن SQL-CE . مابقی مسایل (نکات mapping، کوئری‌ها و غیره) هیچ تفاوتی با سایر بانک‌های اطلاعاتی نخواهد داشت و یکی است. به این معنا که اگر برنامه شما از ویژگی‌های خاص بانک‌های اطلاعاتی استفاده نکند (مثلا اگر از رویه‌های ذخیره شده اس کیوال سرور استفاده نکرده باشد)، فقط با تغییر کانکشن استرینگ و معرفی dialect و driver جدید، به سادگی می‌تواند به یک بانک اطلاعاتی دیگر سوئیچ کند؛ بدون اینکه حتی بخواهید یک سطر از کدهای اصلی برنامه خود را تغییر دهید.

دریافت یک مثال کامل NHibernate 3.2 در این زمینه

تنها نکته جدید آن این متد است:

```
private Configuration getConfig()
{
            var configure = new Configuration();
            configure.SessionFactoryName("BuildIt");

            configure.DataBaseIntegration(db =>
            {
                db.ConnectionProvider<DriverConnectionProvider>();
                db.Dialect<MsSqlCe40Dialect>();
                db.Driver<SqlServerCeDriver>();
                db.KeywordsAutoImport = Hbm2DDLKeyWords.AutoQuote;
                db.IsolationLevel = IsolationLevel.ReadCommitted;
                db.ConnectionString = ConnectionString;
                db.Timeout = 10;

                //for testing ...
                db.LogFormattedSql = true;
                db.LogSqlInConsole = true;
            });

            return configure;
}
```

که در آن نحوه تعریف MsSqlCe40Dialect و SqlServerCeDriver مشخص شده است.

نکته حاشیه‌ای!

در این مثال primary key از نوع identity تعریف شده و بدون مشکل کار کرد. همین را اگر با EF تست کنید، این خطا را دریافت می‌کنید: «Server-generated keys and server-generated values are not supported by SQL Server Compact». بله، EF نمی‌تواند با primary key از نوع identity حین کار با SQL-CE کار کند. برای رفع آن توصیه شده است که از Guid استفاده کنید!

نکته تکمیلی:

استفاده از Dialect سفارشی در NHibernate

نکته پایانی!

و در پایان باید اشاره کرد که SQL-CE یک بانک اطلاعاتی نوشته شده با دات نت نیست (با CPP نوشته شده است و نصب آن هم نیاز به ران تایم به روز VC را دارد). به این معنا که جهت سیستم‌های 64 بیتی و 32 بیتی باید نسخه مناسب آن‌را توزیع کنید. یا اینکه Target platform پروژه جاری دات نت خود را بر روی X86 قرار دهید (نه بر روی Any CPU پیش فرض) و در این حالت تنها یک نسخه X86 بانک اطلاعاتی SQL-CE و همچنین برنامه خود را برای تمام سیستم‌ها توزیع کنید.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۲/۲۷ ۱۷:۴۶:۴۱

جهت تکمیل این مطلب، MsSqlCe40Dialect پیش فرض تعریف یک سری از توابع SQL-CE را ندارد. این کلاس رو تکمیل کردم که از اینجا می‌تونید دریافت کنید: (^)
استفاده از آن هم بسیار ساده است. در متد getConfig فوق، بجای MsSqlCe40Dialect بنویسید CustomMsSqlCe40Dialect

نویسنده: MehdiPayervand
تاریخ: ۱۳۹۰/۱۲/۲۷ ۱۸:۰۸:۲۷

بابت اشتراک مطالبتون ممنون، مهندس نمیدونم امسال وبلاگ چه هدیه ای برای کاربرا داره ;)
سال جدید رو هم بهتون تبریک میگم و آرزوی سالی پر از موفقیت دارم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۲/۲۷ ۲۰:۲۹:۲۱

سلامت باشید؛ سال نوی شما هم مبارک.

عنوان: SQL Server CE و ثبت متون طولانی در EF Code first
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۲ ۱۴:۲۸
آدرس: www.dotnettips.info
برچسب‌ها: Entity framework, SQL Server CE

زمانیکه در EF Code first تعریف خاصیتی به نحو زیر باشد

```
public string Content { get; set; }
```

در حین کار با SQL Server به صورت خودکار به nvarchar max نگاشت می‌شود. اما همین تعریف در SQL Server CE به nvarchar 4000 نگاشت خواهد شد؛ چون این بانک اطلاعاتی نوع‌های max دار را پشتیبانی نمی‌کند.
بنابراین اگر هدف، ثبت اطلاعات در فیلدی از نوع ntext در این بانک اطلاعاتی باشد باید به یکی از دو روش زیر عمل کرد:

```
[MaxLength]
public string Content { get; set; }
```

بله. فقط کافی است یک MaxLength را بالای خاصیت قرار داد (بدون تعیین طول آن) تا به صورت خودکار در SQL Server CE به ntext نگاشت شود و یا می‌توان نوع ستون را صریحا تعیین کرد:

```
[Column(TypeName = "ntext")]
public string Text { get; set; }
```

روش اول بهتر است از این جهت که با بانک‌های اطلاعاتی مختلف سازگاری بهتری دارد. برای مثال نوع ntext در SQL Server کامل، منسوخ شده درنظر گرفته می‌شود اما اگر از ویژگی MaxLength در اینجا استفاده گردد به صورت خودکار به nvarchar max نگاشت خواهد شد و در SQL Server CE به ntext .
بنابراین قید MaxLength بر روی خواصی که قرار است حاوی متونی طولانی باشند، می‌تواند به عنوان یک کار مفید جهت سازگاری با بانک‌های مختلف، به شمار آید.

عنوان: **بروز رسانی استفاده از SqlServer Compact در Entityframework 6.0**
نویسنده: مهدی پایروند
تاریخ: ۱۳۹۲/۰۸/۱۵ ۱۰:۵
آدرس: www.dotnettips.info
برچسب‌ها: Entity framework, NuGet, SQL Server CE, Entity Framework 6, Update-Package

تغییراتی در Entity framework 6 صورت گرفته و در ذیل لیستی از موارد آن آمده است. همچنین پیشتر در همین سایت نیز به آن‌ها اشاره‌ای شده که باز تولید پروایدرها برای نسخه جدید Entity framework یکی از آن‌ها می‌باشد:

```
Rebuilding EF Providers for EF6
Updating Applications to use EF6
EF Tools: adding EF6 support & enabling out-of-band releases
Async Query and Save
Connection Resiliency
Code-Based Configuration
Dependency Resolution
Interception/SQL logging
Custom implementations of Equals or GetHashCode on entity classes
Custom Code First Conventions
Code First Mapping to Insert/Update/Delete Stored Procedures
Configurable Migrations History Table
Multiple Contexts per Database
```

اکنون برای به‌روز رسانی به نسخه جدید، جهت ادامه استفاده از SqlServer Compact مواردی باید لحاظ شود که به آنها اشاره خواهیم کرد و قبل از آنها رعایت یک سری از پیشنیازها لازم است. برای مثال در وب کانفیگ برای استفاده از پروایدر SqlServer Compact بعنوان پروایدر پیش فرض باید قسمت مربوطه را به نحو ذیل تغییر داده باشیم:

```
<entityFramework>
    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlCeConnectionFactory,
EntityFramework">
      <parameters>
        <parameter value="System.Data.SqlServerCe.4.0" />
      </parameters>
    </defaultConnectionFactory>
    <providers>
      <provider invariantName="System.Data.SqlServerCe.4.0"
type="System.Data.Entity.SqlServerCompact.SqlCeProviderServices, EntityFramework.SqlServerCompact" />
    </providers>
  </entityFramework>
```

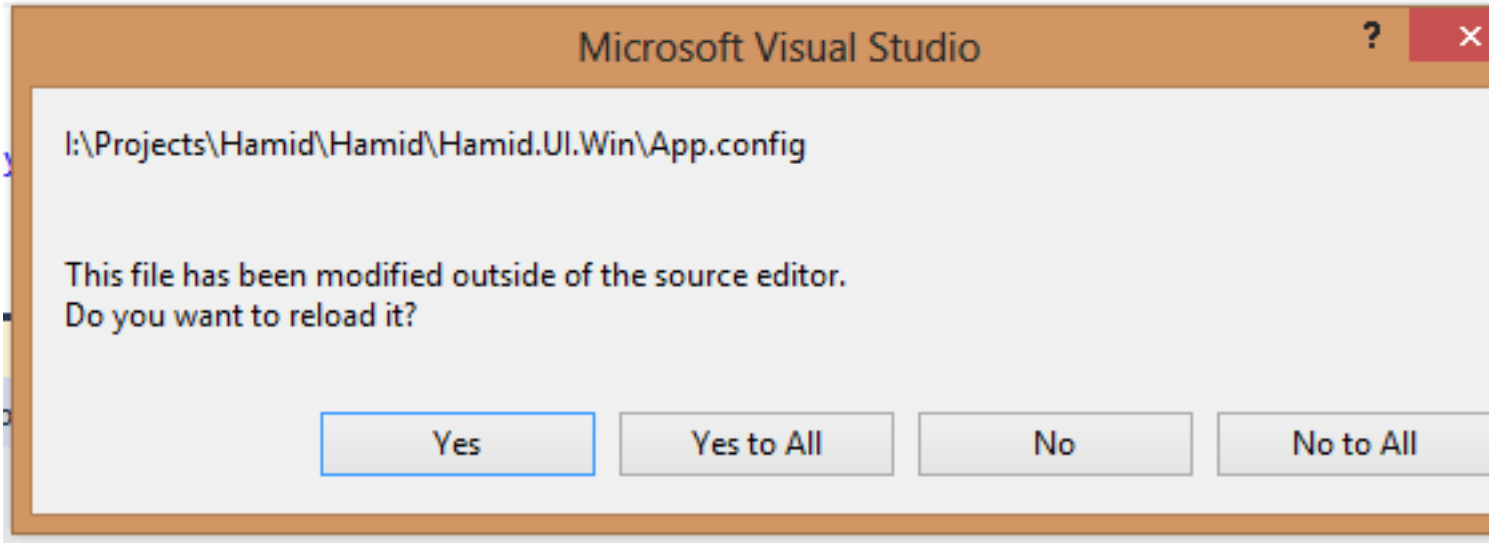
حالا در کنسول نیوگت دستور زیر را برای به‌روزرسانی فقط Entity Framework وارد و اجرا میکنیم:

```
Update-Package EntityFramework
```

پیغام موفقیت آمیز بودن به‌روز رسانی در خروجی نیوگت ظاهر می‌شود

```
PM> Update-Package EntityFramework
Updating 'EntityFramework' from version '5.0.0' to '6.0.1' in project 'Hamid.Core.Model'.
Removing 'EntityFramework 5.0.0' from Hamid.Core.Model.
Successfully removed 'EntityFramework 5.0.0' from Hamid.Core.Model.
```

و نیز تاییدی برای اعمال تغییرات به‌روز رسانی Entity framework انجام میشود تا فایل کانفیگ پروژه را تغییر دهد:



این تغییرات شامل موارد ذیل می‌باشند (در صورت به‌روز رسانی دستی، منظور کپی پکیج بصورت دستی، اعمال تغییرات در کانفیگ‌ها مورد نیاز است):

```
<!-- 1. Change in <configuration><configSections> -->
    <section name="entityFramework"
type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=4.4.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
    <section name="entityFramework"
type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
<!-- 2. Add in <entityFramework><providers> -->
      <provider invariantName="System.Data.SqlClient"
type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
```

بعد از به‌روز رسانی Entityframework باید پکیج EntityFramework.SqlServerCompact برای ادامه استفاده از پروایدر نصب شود که با دستور نیوگت زیر این امر نیز میسر است:

```
PM> Install-Package EntityFramework.SqlServerCompact
```

حالا بدون مشکل می‌توان از پروژه بیلد گرفت و کار توسعه را ادامه داد.

عنوان: **معرفی نسخه غیر رسمی SSCEVSTools for Visual Studio 2013**
نویسنده: مجتبی کاویانی
تاریخ: ۱۳۹۲/۰۹/۰۱ ۱۶:۱۰
آدرس: www.dotnettips.info
برچسب‌ها: SQL Server CE, Visual Studio 2013

با هر بار عرضه‌ی نسخه‌های جدید ویژوال استادیو، علاوه بر اضافه شدن امکانات جدید، برخی از امکانات هم به دلایل نامعلومی از این نرم افزار حذف می‌شوند. در Visual Studio 2012 امکان بسیار کارآمد Setup and Deployment حذف گردید و این بار برخلاف انتظار در Visual Studio 2013 با عدم پشتیبانی از Sql Server Compact مواجه شدیم و هنور دلایل این کار از سوی تیم ویژوال استادیو توضیح داده نشده است. شاید مایکروسافت در حال توسعه نسخه NoSql جدیدی برای جایگزینی باشد.

می توانید از ابزار SQL Server Compact Toolbox استفاده نمایید که کارایی خوبی ندارد و بیشتر یک مکمل است. ا ما راهی برای بازگشت این ابزار به Visual Studio 2013 وجود دارد؟

قابلیت Data Designer Extensibility

در نگارش‌های مختلف ویژوال استادیو امکانی به نام DDEX Provider وجود دارد که توسط آن می‌توانید یک Data Designer جدید را به ویژوال استادیو اضافه نمایید. در واقع اگر از پنجره Server Explorer بر روی Data Connections راست کلیک و یک کانکشن جدید بسازید، لیست Data Sourceهای پیش فرض ویژوال استادیو به شما نشان داده می‌شود که به کمک همین قابلیت DDEX به ویژوال استادیو اضافه شده است. با این قابلیت، امکان اضافه نمون یک Data Designer برای یک پایگاه داده نیز وجود دارد. از آدرس Data Designer Extensibility (DDEX) SDK می توانید نحوه تولید و رجیستر کردن یک DDEX Provider را بیاموزید. برای مثال رجیستری زیر IBM DB2 Data Provider را به ویژوال استادیو اضافه می‌نماید

```
HKLM
{
   %REGROOTBEGIN%

   'DataProviders'
   {
      '{6085DDE2-2EE1-4768-82C3-5425D9B98DAD}' = s 'IBM DB2 Provider'
      {
         val 'DisplayName' = s 'Provider_DisplayName, IBM.DB2.Resources'
         val 'ShortDisplayName' = s 'Provider_ShortDisplayName, IBM.DB2.Resources'
         val 'Description' = s 'Provider_Description, IBM.DB2.Resources'
         val 'FactoryService'= s'{45E1413D-896C-4a2a-A75C-1CBCA51C80CB}'
         val 'Technology' = s '{6565551F-A496-45f3-AFFB-D1AECA082824}'
         val 'InvariantName' = s 'IBM.DB2'
         val 'PlatformVersion' = s '2.0'

         'SupportedObjects'
         {
            'IVsDataViewSupport'
            'IVsDataObjectSupport'
            'IVsDataConnectionUIControl'
            'IVsDataConnectionProperties'
            'IVsDataConnectionSupport'
         }
      }
   }

   'Services'
   {
      '{45E1413D-896C-4a2a-A75C-1CBCA51C80CB}' = s '{7B7F1923-D8F9-430f-9FA7-7919677E5EAC}'
      {
         val 'Name' = 'IBM DB2 Provider Object Factory'
      }
   }

   'Packages'
   {
      '{7B7F1923-D8F9-430f-9FA7-7919677E5EAC}' = 'DB2 Package'
      {
         val 'InProcServer32' = s 'mscoree.dll'
         val 'Class' = s 'IBM.DB2.DB2Package'
         val 'Codebase' = s '%MODULE%'

         'SatelliteDll'
         {
             val 'Path' = s '%PATH%'
```

```
            val 'DllName' = s 'IBM.DB2UI.DLL'
        }
    }

    %REGROOTEND%
}
```

ابزار SSCEVSTools for Visual Studio 2013

برای اضافه نمودن Sql Server Compact Data Provider به Visual Studio 2013 از نسخه قبلی SSCEVSTools که برای Visual Studio 2012 عرضه شده است استفاده می‌کنیم. در واقع این ابزار یک DDEX Provider را به ویژوال استادیو برای Sql Server Compact اضافه می‌کند. اما این نصب کننده، برای نسخه‌ی قبل، تهیه شده است و امکان نصب آن بر روی Visual Studio 2013 نمی‌باشد. یک راهکار عملی، دسترسی به فایل‌ها و رجیستری‌های موجود در این نصب کننده و تولید نصب کننده جدیدی می‌باشد.

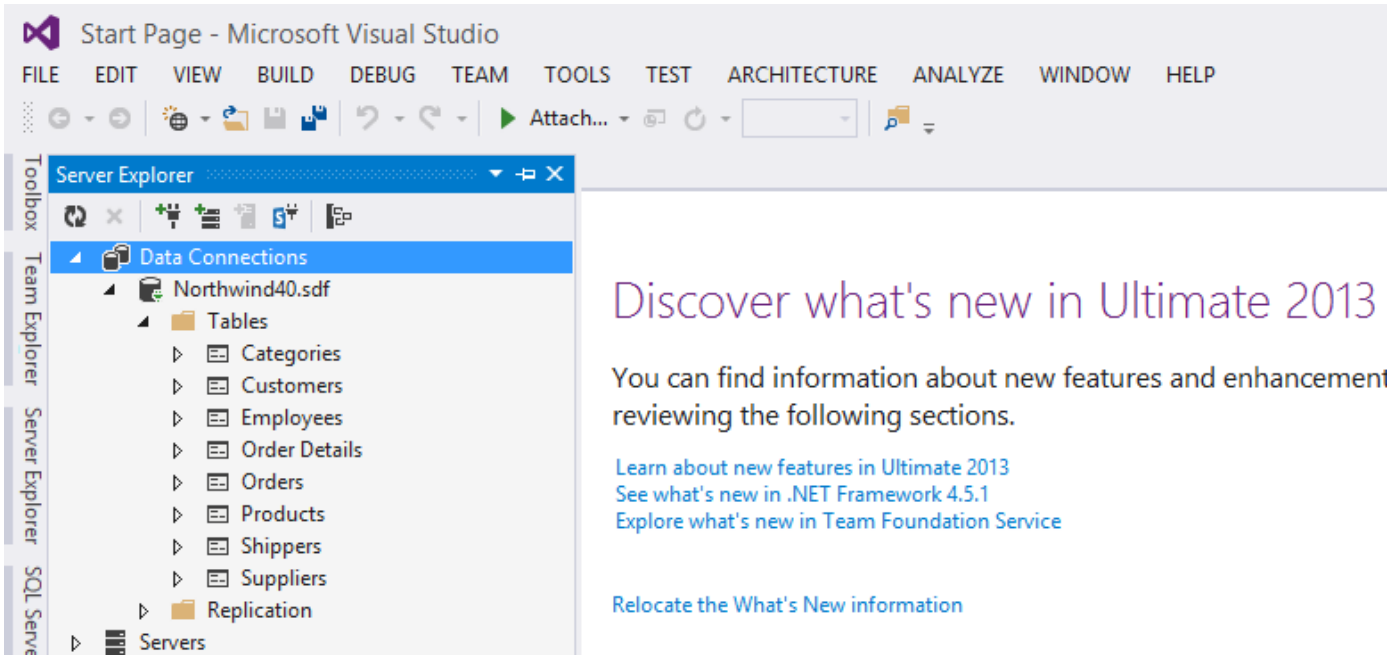
دسترسی به محتوی فایل‌های Setup

ابزار Orca در Windows SDK برای ویرایش فایل‌های نصب کننده توسط مایکروسافت تولید شده است که امکان مشاهده تمامی جزئیات آن را فراهم می‌نماید. ابزار قبلی، شامل فایل‌های dll و رجیستری است و امکان اتصال به Sql Server Compact را به ویژوال استادیو اضافه می‌نمود.

حال با یک برنامه Setup ساز، فایل‌ها و رجیستری را برای Visual Studio 2013 تنظیم نموده و با نصب ابزار جدید، دوباره امکان استفاده از Sql Server Compact در Visual Studio 2013 میسر می‌شود.

برای نصب این ابزار، آن را از گالری ویژوال استادیو به نام SSCEVSTools for Visual Studio 2013 دانلود نمایید.

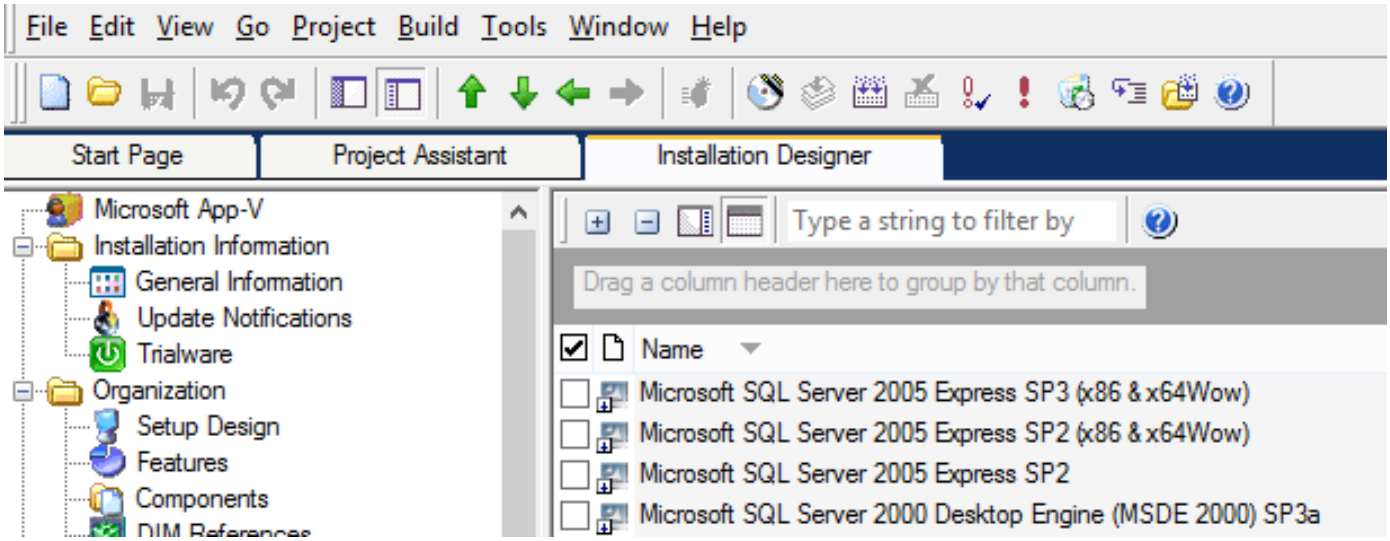
البته چون این ابزار بصورت غیر رسمی تولید و عرضه شده است گاهی اوقات به صورت خودکار از لیست Data Sourceها حذف شده که لازم است آن را حذف و مجددا نصب نمایید.



اگر مایل به بازگشت و کار بر روی نسخه جدید Sql Server Compact 5 هستید اینجا در Visual Studio UserVoice رای دهید.

عنوان: **افزودن SQL Server CE 4.0 به لیست توزیع مجدد در InstallShield 2010**
نویسنده: مهدی پایروند
تاریخ: ۱۳۹۲/۱۰/۲۵ ۲۱:۳۵
آدرس: www.dotnettips.info
گروه‌ها: SQL Server CE, Software deployment, Embedded Database, InstallShield, InstallShield-SetupPrerequisites

در برنامه‌ی ساخت نصاب InstallShield، در قسمت افزودن بسته‌های نصبی برای برنامه‌ی ساخته شده



بسته‌ی نصب SQL Server CE 3.5 SP2 وجود دارد:

اما برای برنامه‌های جدیدتر نیاز به افزودن بسته‌ی نصب دیتابیس SQL Server CE نسخه 4 است که با عدم وجود این بسته روبرو هستیم. در ادامه با نحوه‌ی افزودن این بسته‌ها آشنا خواهید شد.

اینگونه بسته‌ها در کنار برنامه‌ی ساخت نصاب و در پوشه‌ی SetupPrerequisites نگهداری شده و با نوع *.prq ذخیره می‌شوند. این نوع فایل‌ها از نوع xml هستند و در واقع یک نوع کار نگاشت را انجام می‌دهند. برای نمونه محتویات یکی از این فایل‌ها را در زیر می‌بینید:

```
<?xml version="1.0" encoding="UTF-8"?>
<SetupPrereq>
<conditions>
<condition Type="32" Comparison="2" Path="HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server
Compact Edition\v3.5\ENU" FileName="DesktopRuntimeVersion" ReturnValue="3.5.8080.0"></condition>
</conditions>
<operatingsystemconditions>
<operatingsystemcondition MajorVersion="5" MinorVersion="0" PlatformId="2" CSDVersion=""
ServicePackMajorMin="3"></operatingsystemcondition>
<operatingsystemcondition MajorVersion="5" MinorVersion="1" PlatformId="2" CSDVersion="" Bits="1"
ProductType="1"></operatingsystemcondition>
<operatingsystemcondition MajorVersion="6" MinorVersion="0" PlatformId="2" CSDVersion=""
Bits="1"></operatingsystemcondition>
<operatingsystemcondition MajorVersion="5" MinorVersion="2" PlatformId="2" CSDVersion="" Bits="1"
ProductType="2|3"></operatingsystemcondition>
<operatingsystemcondition MajorVersion="6" MinorVersion="1" PlatformId="2" CSDVersion=""
```

```
Bits="1"></operatingsystemcondition>
<operatingsystemcondition MajorVersion="6" MinorVersion="0" PlatformId="2" CSDVersion="" Bits="1"
ProductType="2|3"></operatingsystemcondition>
</operatingsystemconditions>
<files>
<file LocalFile="&lt;ISProductFolder&gt;\SetupPrerequisites\SQL CE 3.5\SSCERuntime_x86-ENU.msi"
URL="http://go.microsoft.com/fwlink/?LinkId=166085&amp;clcid=0x409"
CheckSum="86AF6D36DFF214718DCD35D851249D3D" FileSize="0,3164160"></file>
</files>
<execute file="SSCERuntime_x86-ENU.msi" cmdline="/q /norestart" cmdlinesilent="/q /norestart"
returncodetoreboot="1641,3010,4123" requiresmsiengine="1"></execute>
<properties Id="{A7C4B3C0-F3A0-426A-A043-E13DBA123E52}" Description="This prerequisite installs the
Microsoft SQL Server Compact 3.5 SP2."
AltPrqURL="http://saturn.installshield.com/is/prerequisites/microsoft sql ce 3.5 sp2.prq"></properties>
<behavior Reboot="2"></behavior>
</SetupPrereq>
```

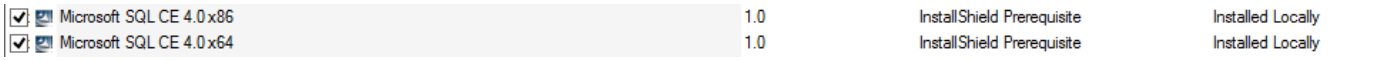
کافیست به ازای هر نسخه‌ی 32 و یا 64 بیتی، فایل xml مورد نظر، با پسوند prq در پوشه‌ی SetupPrerequisites ذخیره شود.
برای نسخه 32 بیتی(Microsoft SQL CE 4.0 x86.prq):

```
<?xml version="1.0" encoding="UTF-8"?>
<SetupPrereq>
    <conditions>
        <condition Type="32" Comparison="2" Path="HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server Compact Edition\v4.0\ENU" FileName="DesktopRuntimeVersion" ReturnValue="4.0.8482.1"></condition>
    </conditions>
    <operatingsystemconditions>
        <operatingsystemcondition CSDVersion="" Bits="1"></operatingsystemcondition>
    </operatingsystemconditions>
    <files>
        <file LocalFile=".\SSCERuntime_x86-ENU.exe"
URL="http://download.microsoft.com/download/0/5/D/05DCCDB5-57E0-4314-A016-874F228A8FAD/SSCERuntime_x86-
ENU.exe" CheckSum="0A55733CF406FBD05DFCFF5A27A0B4F7" FileSize="0,2379544"></file>
    </files>
    <execute file="SSCERuntime_x86-ENU.exe"></execute>
    <properties Id="{2754916B-119B-4428-9F94-DC9E45072CCC}"></properties>
    <behavior Failure="4" Reboot="2"></behavior>
</SetupPrereq>
```

و برای نسخه 64 بیتی(Microsoft SQL CE 4.0 x64.prq) :

```
<?xml version="1.0" encoding="UTF-8"?>
<SetupPrereq>
    <conditions>
        <condition Type="32" Comparison="2" Path="HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server Compact Edition\v4.0\ENU" FileName="DesktopRuntimeVersion" ReturnValue="4.0.8482.1"></condition>
    </conditions>
    <operatingsystemconditions>
        <operatingsystemcondition CSDVersion="" Bits="2"></operatingsystemcondition>
    </operatingsystemconditions>
    <files>
        <file LocalFile=".\SSCERuntime_x64-ENU.exe"
URL="http://download.microsoft.com/download/0/5/D/05DCCDB5-57E0-4314-A016-874F228A8FAD/SSCERuntime_x64-
ENU.exe" CheckSum="A417082ECAEDD95AFB41F73DC140C350" FileSize="0,2621240"></file>
    </files>
    <execute file="SSCERuntime_x64-ENU.exe"></execute>
    <properties Id="{7CB7BE3C-614A-403F-94D9-5652285A3EDF}"></properties>
    <behavior Failure="4" Reboot="2"></behavior>
</SetupPrereq>
```

و در نهایت این دو بسته به لیست اضافه خواهد شد:



نظرات خوانندگان

نویسنده: ایمان رضایی پور
تاریخ: ۱۳۹۲/۱۰/۲۷ ۱۱:۴۶

در اینجا بیشترین قابلیت‌ها را SQL CE 3.5 SP2 به خود اختصاص داده است. به نظر شما دلیل آن چیست؟
مگر SQL CE 4 جدیدتر نیست؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۲۷ ۱۳:۱

Features not supported in SQL Server Compact 4.0

علتش این است که مثلا LINQ to SQL دیگر با SQL CE 4 پشتیبانی نمی‌شود چون خود LINQ to SQL دیگر توسط MS توسعه جدی پیدا نمی‌کند و به EF سوئیچ شده و EF هم پروایدر رسمی برای SQL CE 4 داره.

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۲/۱۰/۲۸ ۸:۵۲

یکی دیگه از دلایلی که بیشتر از نسخه 3.5 صحبت میشه میتونه این باشه که نهایت نسخه ای است که در سری Windows CE و خصوصا 6.0 استفاده میشه. البته آخرین نسخه SQL Server Compact 3.5 SP2 هستش.