Using SQLite with the Entity Framework 6 designer - a troubleshooting guide

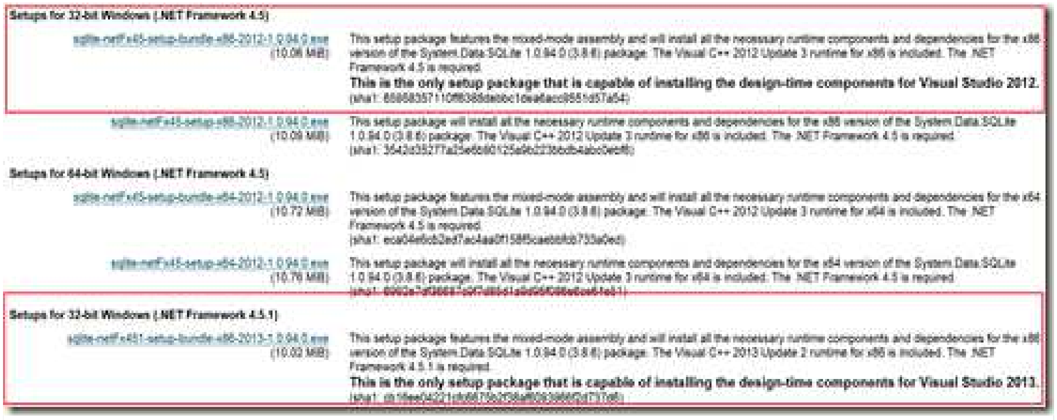
Many users would like to be able to use the SQLite database with the Entity Framework 6 EDM Wizard and Designer, but looking at Stack Overflow, they often face problems attempting to do that. This blog post describes the steps required to do that, and gives some troubleshooting advice and tips.

Install the SQLite DDEX provider

In order to use the Entity Data Model Wizard to generate a Code First or EDMX Model from an existing database, you must be able to connect to the database from Server Explorer in Visual Studio. In order to do that, a so called DDEX provider for the database and Visual Studio version in use must be installed.

For SQLite you must download the provider fromhttp://system.data.sqlite.org/index.html/doc/trunk/www/downloads.wiki - this page is an unreadable mess, sadly.

For Visual Studio 2013, the download you need is named: sqlite-netFx451-setup-bundle-x86-2013-1.0.94.0.exe for the current version of the ADO.NET provider, so 1.0.94 might change to 1.0.95 or higher in the future.



Setups for 32-bit Windows (.NET Framework 4.5)

sqlite-netFx45-setup-bundle-x86-2012-1.0.94.0.exe (10.06 MiB)

This setup package features the mixed-mode assembly and will install all the necessary runtime components and dependencies for the x86 version of the System.Data.SQLite 1.0.94.0 (3.8.6) package. The Visual C++ 2012 Update 3 runtime for x86 is included. The .NET Framework 4.5 is required.
This is the only setup package that is capable of installing the design-time components for Visual Studio 2012.
(sha1: 65958357110ff8388dabbc1dea6acc8551d57a54)

sqlite-netFx45-setup-x86-2012-1.0.94.0.exe (10.09 MiB)

This setup package will install all the necessary runtime components and dependencies for the x86 version of the System.Data.SQLite 1.0.94.0 (3.8.6) package. The Visual C++ 2012 Update 3 runtime for x86 is included. The .NET Framework 4.5 is required.
(sha1: 3542d35277a25e6b80125a9b2236bdb4abc0ebf6)

Setups for 64-bit Windows (.NET Framework 4.5)

sqlite-netFx45-setup-bundle-x64-2012-1.0.94.0.exe (10.72 MiB)

This setup package features the mixed-mode assembly and will install all the necessary runtime components and dependencies for the x64 version of the System.Data.SQLite 1.0.94.0 (3.8.6) package. The Visual C++ 2012 Update 3 runtime for x64 is included. The .NET Framework 4.5 is required.
(sha1: eca04e6cb2ed7ac4aa0f158f5cae66fdb733a0ed)

sqlite-netFx45-setup-x64-2012-1.0.94.0.exe (10.76 MiB)

This setup package will install all the necessary runtime components and dependencies for the x64 version of the System.Data.SQLite 1.0.94.0 (3.8.6) package. The Visual C++ 2012 Update 3 runtime for x64 is included. The .NET Framework 4.5 is required.
(sha1: 6992a7df3868 [illegible])

Setups for 32-bit Windows (.NET Framework 4.5.1)

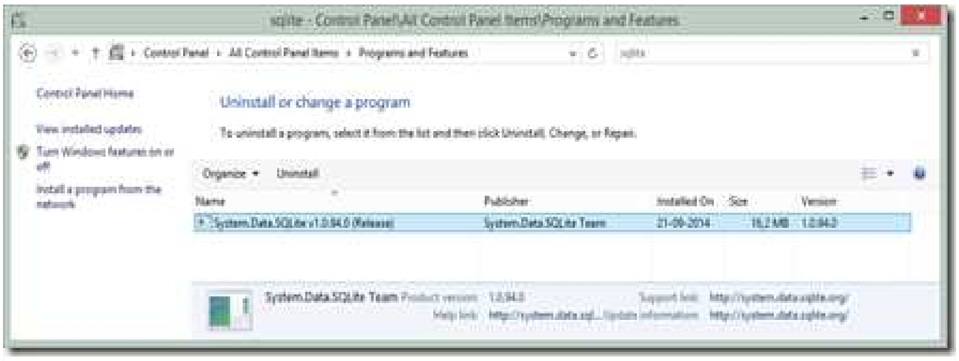
sqlite-netFx451-setup-bundle-x86-2013-1.0.94.0.exe (10.02 MiB)

This setup package features the mixed-mode assembly and will install all the necessary runtime components and dependencies for the x86 version of the System.Data.SQLite 1.0.94.0 (3.8.6) package. The Visual C++ 2013 Update 2 runtime for x86 is included. The .NET Framework 4.5.1 is required.
This is the only setup package that is capable of installing the design-time components for Visual Studio 2013.
(sha1: dc16ee04221dcb6759d2f38af6063986f2d737d6)

It is important that the DDEXprovider version matches the version of the current SQLite EF6 NuGet package that you use in your project!

Make sure to enable Visual Studio integration and GAC registration during installation.

You can check which version of the DDEX provider you have installed in “Add or Remove Programs”:



sqlite - Control Panel\All Control Panel Items\Programs and Features

Control Panel › All Control Panel Items › Programs and Features

sqlite
Control Panel Home
View installed updates
Turn Windows features on or off
Install a program from the network
Organize ▾
Uninstall

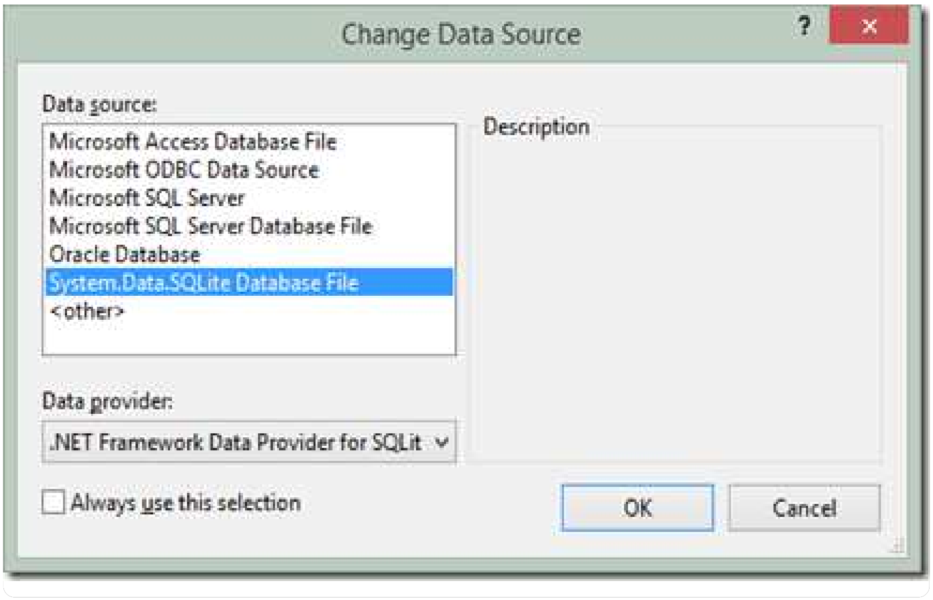
Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

Name	Publisher	Installed On	Size	Version
System.Data.SQLite v1.0.94.0 (Release)	System.Data.SQLite Team	21-09-2014	16,2 MB	1.0.94.0

System.Data.SQLite Team Product version: 1.0.94.0 Support link: http://system.data.sqlite.org/
Help link: http://system.data.sql... Update information: http://system.data.sqlite.org/

Once installed, you can add a connection to your SQLite database from Server Explorer:



Change Data Source

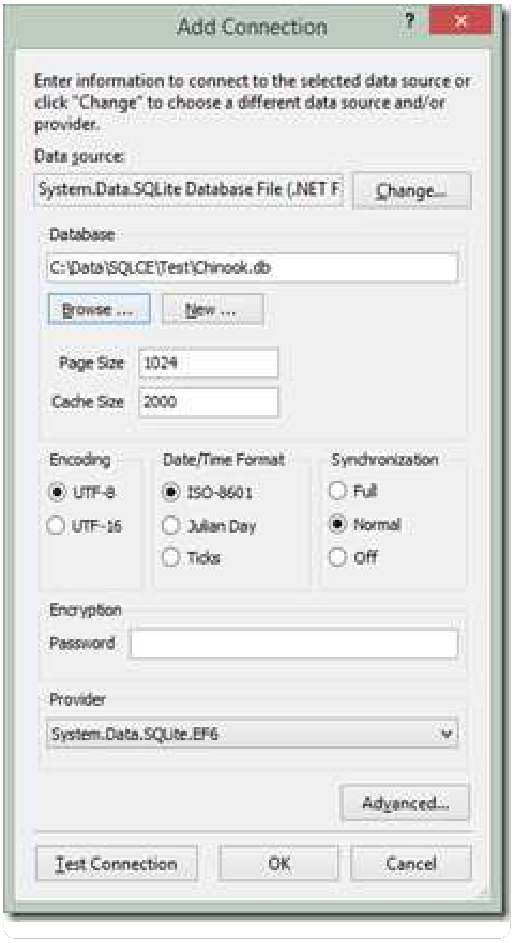
Data source:

- Microsoft Access Database File
- Microsoft ODBC Data Source
- Microsoft SQL Server
- Microsoft SQL Server Database File
- Oracle Database
- System.Data.SQLite Database File
- <other>

Description

Data provider: .NET Framework Data Provider for SQLit

□ Always use this selection



Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: System.Data.SQLite Database File (.NET F

Database: C:\Data\SQLCE\Test\Chinook.db

Page Size: 1024

Cache Size: 2000

Encoding: ● UTF-8 ○ UTF-16

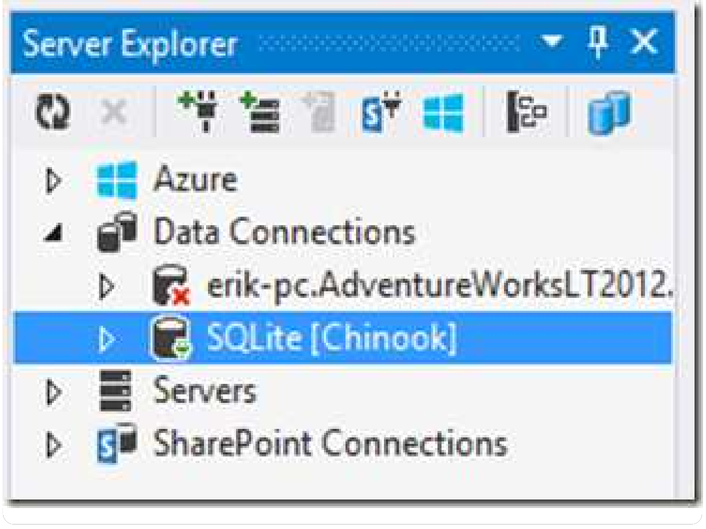
Date/Time Format: ● ISO-8601 ○ Julian Day ○ Ticks

Synchronization: ○ Full ● Normal ○ Off

Encryption

Password:

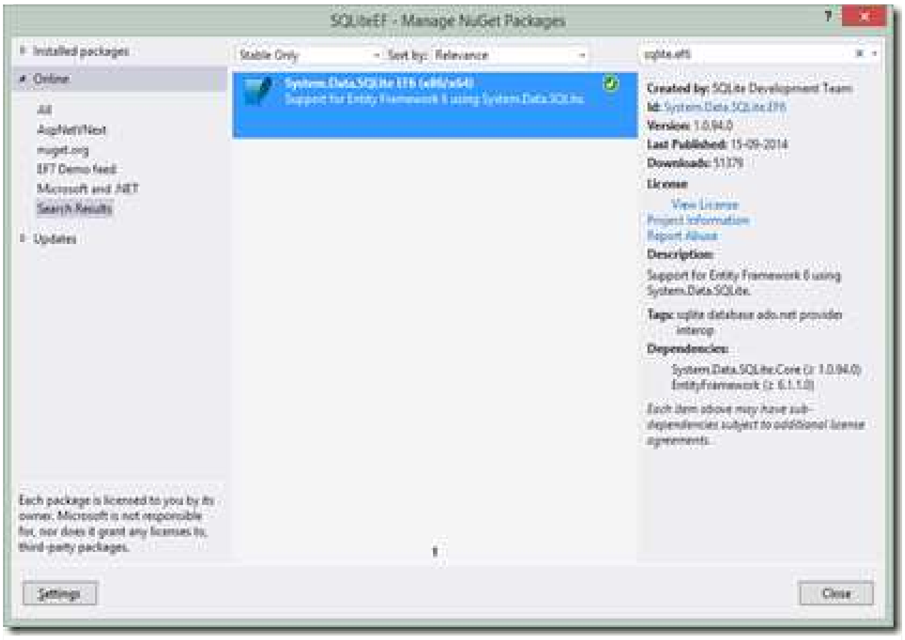
Provider: System.Data.SQLite.EF6



Add the correct SQLite EF6 NuGet package to your project

To demonstrate using the SQLite EF6 provider, let’s go through creating a console app in Visual Studio 2013, that uses the Chinook sample database that we connected to above.

To add the SQLite EF6 provider to your project, select the System.Data.SQLite.EF6 package:



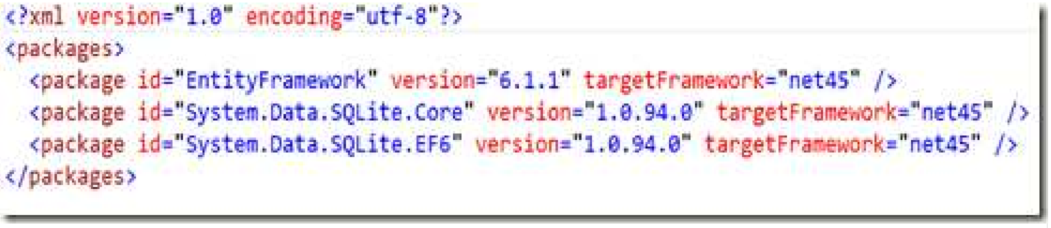
Created by: SQLite Development Team
Id: System.Data.SQLite.EF6
Version: 1.0.94.0
Last Published: 15-09-2014
Downloads: 51379
License
View License
Project Information
Report Abuse
Description:
Support for Entity Framework 6 using System.Data.SQLite.
Tags: sqlite database ado.net provider interop
Dependencies:
System.Data.SQLite.Core (≥ 1.0.94.0)
EntityFramework (≥ 6.1.1.0)

Each item above may have sub-dependencies subject to additional license agreements.

Each package is licensed to you by its owner. Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.

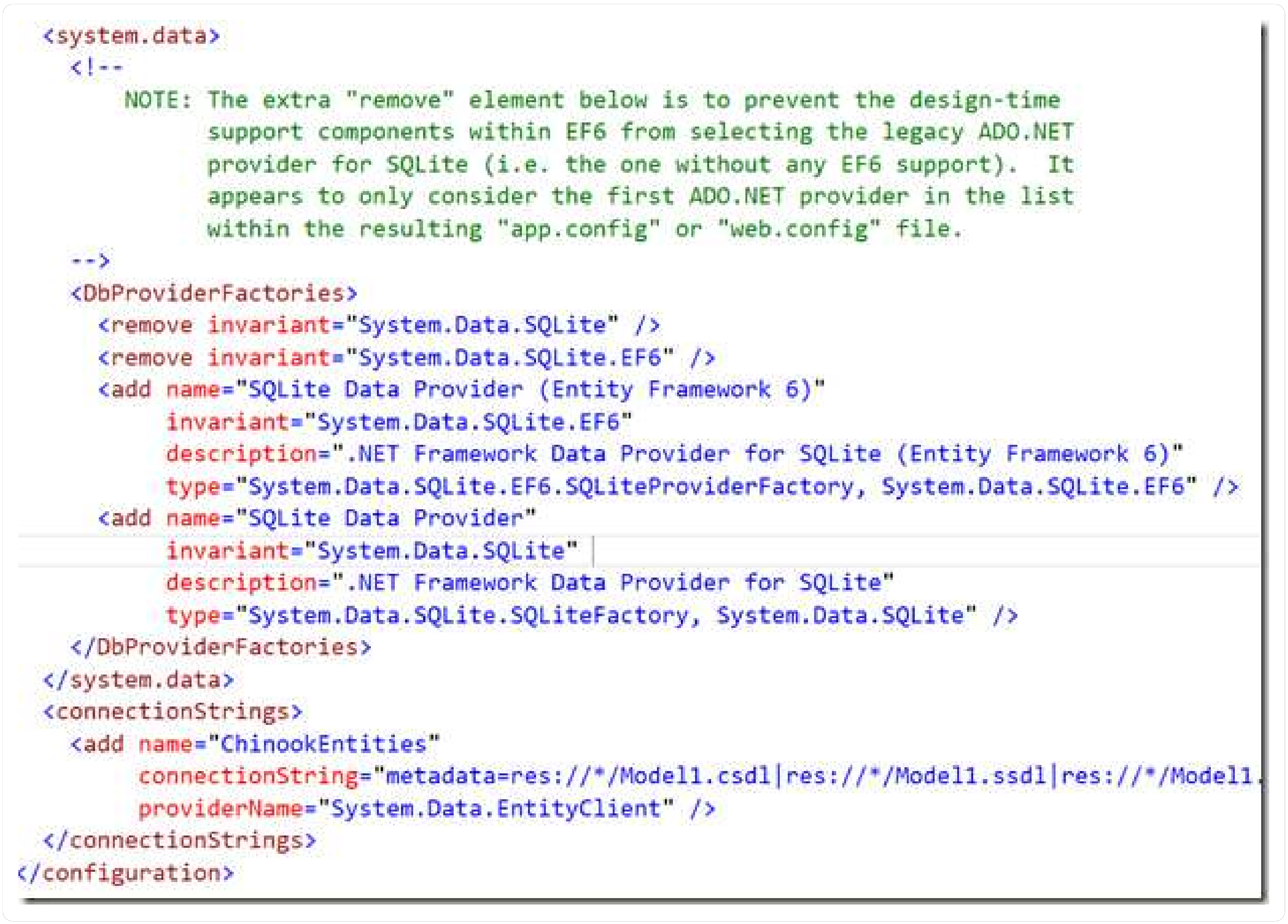
This package will install all other required dependencies, including Entity Framework 6.1.1.

After installation of this package in a project that target .NET 4.5, your packages.config should look like this:



```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="EntityFramework" version="6.1.1" targetFramework="net45" />
  <package id="System.Data.SQLite.Core" version="1.0.94.0" targetFramework="net45" />
  <package id="System.Data.SQLite.EF6" version="1.0.94.0" targetFramework="net45" />
</packages>
```

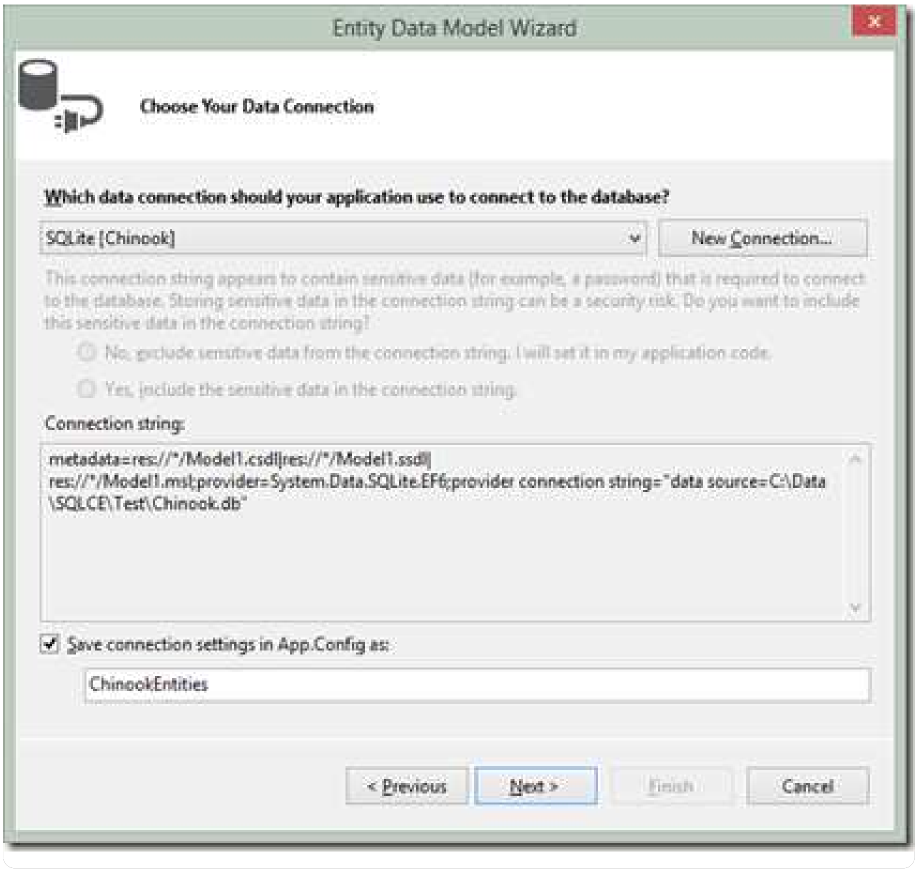
And the system.data section of your app.config file should look like this (**UPDATE:** notice the additional **add** entry added to avoid runtime errors!)



```
  <system.data>
    <!--
        NOTE: The extra "remove" element below is to prevent the design-time
              support components within EF6 from selecting the legacy ADO.NET
              provider for SQLite (i.e. the one without any EF6 support).  It
              appears to only consider the first ADO.NET provider in the list
              within the resulting "app.config" or "web.config" file.
    -->
    <DbProviderFactories>
      <remove invariant="System.Data.SQLite" />
      <remove invariant="System.Data.SQLite.EF6" />
      <add name="SQLite Data Provider (Entity Framework 6)"
           invariant="System.Data.SQLite.EF6"
           description=".NET Framework Data Provider for SQLite (Entity Framework 6)"
           type="System.Data.SQLite.EF6.SQLiteProviderFactory, System.Data.SQLite.EF6" />
      <add name="SQLite Data Provider"
           invariant="System.Data.SQLite"
           description=".NET Framework Data Provider for SQLite"
           type="System.Data.SQLite.SQLiteFactory, System.Data.SQLite" />
    </DbProviderFactories>
  </system.data>
  <connectionStrings>
    <add name="ChinookEntities"
         connectionString="metadata=res://*/Model1.csdl|res://*/Model1.ssdl|res://*/Model1.
         providerName="System.Data.EntityClient" />
  </connectionStrings>
</configuration>
```

Now BUILD your project!

You should now be able to add a EDMX or Code First based model to your project, using Add New Item. Data, ADO.NET Entity Data Model:



Which data connection should your application use to connect to the database?

SQLite [Chinook]

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

○ No, exclude sensitive data from the connection string. I will set it in my application code.

○ Yes, include the sensitive data in the connection string.

Connection string:

```
metadata=res://*/Model1.csdl|res://*/Model1.ssdl|
res://*/Model1.msl;provider=System.Data.SQLite.EF6;provider connection string="data source=C:\Data
\SQLCE\Test\Chinook.db"
```

☑ Save connection settings in App.Config as:

ChinookEntities

Add some code to the Main method to test that the application also works in runtime:

```
            using (var db = new ChinookEntities())
            {
                var albums = db.Albums.ToList();
            }
```

If you are unable to see the connection to the SQLite database we made earlier, here are a couple of tips:

Try building the project and possible changing the build configuration to x86 has worked for me.

Installing the latest Entity Framework 6.1.2 Visual Studio tools have worked very well for me – download link in the blog post here. I think it is due to this fix in the Tools beta.

Hope this helps!

g+1 Recommend this on Google Indsendt af ErikEJkl. 6:45 PM

Etiketter: ADO.NET, Entity Framework, SQLIte

23 comments:

Roy Triesscheijn said...

Even when going through all these steps I still cannot get SQLite to work correctly with EF6.

Whenever I try to add migrations it still seems to be looking for the non-EF6 versions.

"Unable to determine the provider name for provider factory of type 'System.Data.SQLite.SQLiteFactory'. Make sure that the ADO.NET provider is installed or registered in the application config."

While (I believe) it should be using: System.Data.SqLite.EF6.SQLiteProviderFactory.

Any ideas?

November 25, 2014 at 3:11 PM

ErikEJ said...

Please check your machine.config file - DbProviderFactories section, if that turns out to the issue, I will update the blog post. C:\Windows\Microsoft.NET\Framework\v4.0.30319\Config\machine.config - the SQLite Data Provider entry. And I dont think the SQLite provider supports migrations!
http://system.data.sqlite.org/index.html/tktview?name=6b5ef4ff7a

November 25, 2014 at 6:08 PM

Belahcene Benzara Tahar said...

Thanks for the post, I followed exactly what you said but I am having a problem when trying to connect to the sqlite database from the server explorer: here is the error message i got once i validate the database file and press OK:
Unable to add data connection
Could not find any resources appropriate fot the specified culture or the neutral culture.
Make sure
"SQLite.Designer.SQLiteDataViewSupport2013.xml" was correctly embedded or linked into assembly "SQLite.Designer" at compile time, or that all the satellite assemblies reqired are loadable and fully signed.

any ideas?

December 31, 2014 at 3:45 PM

ErikEJ said...

Belachene - looks like a sqlite designer issue. http://system.data.sqlite.org/index.html/tktview/9429dee35bb213db615b13078aeb5c20b4748ca4?plaintext - sure you have the latest designer bits installed, then report at the sqlite site

December 31, 2014 at 4:13 PM

Belahcene Benzara Tahar said...

Thanks Erik;
I've resolved the issue; the reason was that I installed **sqlite-netFx451-setup-bundle-x86-2013-1.0.94.0.exe** in top of **sqlite-netFx45-setup-bundle-x86-2012-1.0.94.0.exe** without unistalling it. I uninstalled it and then reinstalled it and it worked like a charm.
Thanks for the great blog :)

December 31, 2014 at 4:18 PM

impy said...

The 1.0.94.1 sqlite dll is version stamped 1.0.94.0. That's no good. Do I need to assembly.load it cos the "null was returned..." bug still exists.

January 16, 2015 at 2:03 AM

ErikEJ said...

Impy: please post an issue on the Codeplex issue tracker or the sqlite site, I have no idea what your issue is

January 16, 2015 at 9:00 AM

Simon Shaw said...

I went through these steps using VS2013 and everything works fine until I get to the Entity Data Model Wizard where my database is not shown and when I click on the new connection button I don't get an option for SQLite (which is kind of strange as it appeared in the server explored stage earlier.

February 4, 2015 at 7:52 AM

ErikEJ said...

Simon: Did you install the latest EF 6 VS Tools? Does the SQLite versions in GAC match your DDEX provider version?

February 4, 2015 at 8:47 AM

António Albuquerque said...

After doing all of the tutorial, I tried to get data from the database and I get the following error: "Unable to determine the provider name for provider factory of type 'System.Data.SQLite.SQLiteFactory'. Make sure that the ADO.NET provider is installed or registered in the application config."

February 17, 2015 at 4:12 PM

ErikEJ said...

Antonio: It is hard to tell what the issue is based on this error message, suggest you provide more info or even better a repro project on Stackoverflow or MSDN

February 17, 2015 at 6:58 PM

Michel Feinstein said...

Hi Erik, I have followed all your steps, and I can't see the sqlite connection under the Entity Data Model Wizard. I have installed the EF tools, your SQLite Toolbox and compiled it for x86....what else can I do to make this work?

February 20, 2015 at 12:40 AM

ErikEJ said...

Michael: See my reply on StackOverflow

February 20, 2015 at 11:03 AM

António Albuquerque said...

ErikEJ, I finally made it work. I had to configure the DbProviderFactories like this:

```
<DbProviderFactories>
<remove invariant="System.Data.SQLite" />
<remove invariant="System.Data.SQLite.EF6" />
<add name="SQLite Data Provider (Entity Framework 6)" invariant="System.Data.SQLite.EF6"
description=".NET Framework Data Provider for SQLite (Entity Framework 6)"
type="System.Data.SQLite.EF6.SQLiteProviderFactory, System.Data.SQLite.EF6" />
<add name="SQLite Data Provider" invariant="System.Data.SQLite" description=".NET Framework Data
Provider for SQLite" type="System.Data.SQLite.SQLiteFactory, System.Data.SQLite" />
</DbProviderFactories>
```

February 20, 2015 at 11:29 AM

ErikEJ said...

Antonio: Does that work both runtime and designtime?

February 20, 2015 at 11:38 AM

António Albuquerque said...

ErikEJ, yes. I can send you my App.config if you want.

Basically after many botched attemps, I followed your tutorial and my steps were:

1. Uninstall all older Sqlite connectors
2. Install lastest 2013 connector as specified on your post
3. New project but before trying to create the EDMX I must build the project (even if it's empty) or else it will fail the wizard on the last step. This is also true for the MySQL connector.
4. The connection string must have a full path or in design time, the wizard can't find any tables (basically can't find the file). I think there might be a work around this but I didn't wasted any time checking
5. All done and I got that error that I reported. After some google time and trial and error I realized that you had to add the second DbFactory for the invariant name System.Data.SQLite. I don't know if it uses the other factory for design time and this one for runtime.

Anyway it works. I hope my steps could help others struggling to get EF working with Sqlite.