# EF适用于各种数据库

## 简介

本文讨论将Entity Framework的Code First应用于各种类型的数据库的情况。目前文档讨论的数据库有SQL Server、My Sql、Sqlite。

## 将EF应用于Sql Server

##### 连接字符串

<!--连接字符串

Server: 需要连接的服务器

DataBase: 连接的数据库

User Id:数据库账号

Password:数据库密码

-->

<add name="ConnString" connectionString="Server=CMLU-PC\CMLU;Database=TestDb;User Id=sa;Password=558276344;" providerName="System.Data.SqlClient" />

IsRowVersion暂时只适用于SQL Server,其它的数据库需百度其它解决方案。

## 将EF应用于My Sql

##### 文档

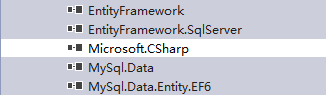
http://dev.mysql.com/doc/connector-net/en/connector-net-introduction.html

##### 开发指南

1. 通过NuGet添加MySql.Data.Entity



添加之后程序中出现



1. 配置连接字符串

<connectionStrings>

<add name="MySqlConStr" connectionString="server=localhost;port=3306;database=DbTest;uid=root;password=558276344" providerName="MySql.Data.MySqlClient"/>

</connectionStrings>

3）使用long类型作为主键，使用guid作为主键时，数据库对应Char(36).

## 将EF应用于SQLite

##### SQLite特征

SQLite无配置，无服务器，单一数据库文件，跨平台.

SQLite是弱类型的，类型通过值来推断。实际上类型和值时合并起来存储的。SQLite变长记录，字段不需要指定长度。

Sqlite支持如下数据类型：

NULL: the value is a NULL value.

INTEGER: The value is a signed integer

REAL: The value is a floating point value, stored as an 8-byte IEEE floating point number.

TEXT: The value is a text string, stored using the database encoding (UTF-8 UTF-16BE UTF-16LE)

BLOB: 一个数据块，完全按照输入存放

参考文档：

http://system.data.sqlite.org/index.html/doc/trunk/www/index.wiki

##### 开发指南

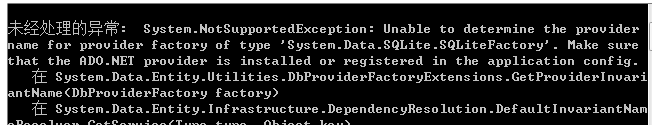
1. 通过NuGet添加System.Data.SQLite



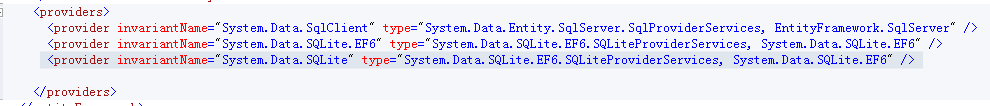
1. 添加连接字符串

<add name="SqliteConStr" connectionString="Data Source=.\mydb.db;Version=3;UseUTF8Encoding=True;" providerName="System.Data.SQLite.EF6"/>

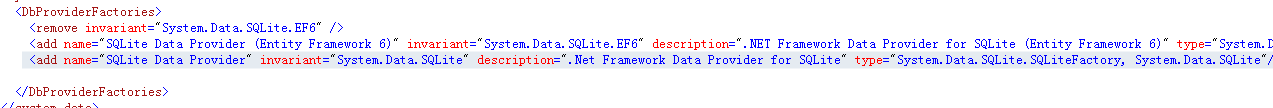
1. 发生如下异常：



在配置文件中两行



<provider invariantName="System.Data.SQLite" type="System.Data.SQLite.EF6.SQLiteProviderServices, System.Data.SQLite.EF6" />



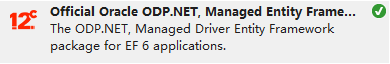
<add name="SQLite Data Provider" invariant="System.Data.SQLite" description=".Net Framework Data Provider for SQLite" type="System.Data.SQLite.SQLiteFactory, System.Data.SQLite"/>

1. 支持Code First

目前SQLite EF不支持Code First,目前使用策略是使用SqliteHelper，先手动创建数据库。再使用EF来兼容数据库。当SQLiteConnection调用Open的时候如果数据库不存在（通过File.Exists检查），则自动使用连接字符串创建数据库

## 将EF应用于Oracle

1）通过NuGet安装Oracle.ManagedDataAccess.EntityFramework



2) Open the app/web.config file to configure the ODP.NET connection string。实际上NuGet已经加了

<oracle.manageddataaccess.client>

<version number="\*">

<dataSources>

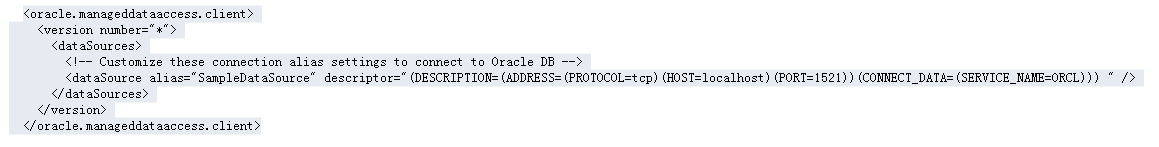
<!-- Customize these connection alias settings to connect to Oracle DB -->

<dataSource alias="SampleDataSource" descriptor="(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=localhost)(PORT=1521))(CONNECT\_DATA=(SERVICE\_NAME=ORCL))) " />

</dataSources>

</version>

</oracle.manageddataaccess.client>



hostname: 主机名，填写服务器所在的IP，如果是本机填localhost

port: 服务器的端口号

service\_name: 数据库实例名

这里指定后就可以忽略tnsnames.ora

tnsnames.ora同时存在于数据库客户端与服务器端。如果在客户端连远程的数据库服务器，你就要编辑客户机上的tnsnames.ora。数据库服务端主机上的tnsnames.ora则可以不编辑。客户端通过tnsnames.ora的配置连接数据库

The <version number=”\*” > tag specifies that all ODP.NET, Managed Driver versions will use these configuration settings.

The <dataSources> tag contains data source aliases and connect descriptor information that were previously in the tnsnames.ora file.

3）Modify the app/web.config file's connection string to create a DbContext your Entity Framework application will use.

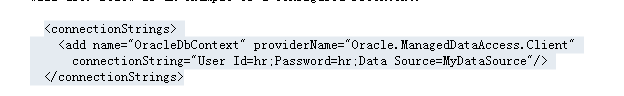
<connectionStrings>

<add name="OracleDbContext" providerName="Oracle.ManagedDataAccess.Client"

connectionString="User Id=hr;Password=hr;Data Source=MyDataSource"/>

</connectionStrings>

实际上NUGET也帮你添加了



DataSource与上面指定的名称一致，这样就不需要tnsnames.ora了