使用注意解决问题：

（1） FDB 中如何处理大小写？好像所有的字母都被搞成了大写？

（2） FDB 中的所有字段似乎都只能用英文！而且还是小写的！

是否跟字符集设定有关？？？

2010-6-13 中文路径问题解决

参考：http://www.jocw.cn/sxsoftservice/Article_Print.asp?ArticleID=780

【Firebird】解决Firebird的.Net Provider中文路径问题

作者：佚名 转贴自：本站原创

这几天试用了一下Firebird数据库，2.0 + .Net Provider，嵌入式访问。果然遇到了传闻已久的“中文路径无法访问”问题，凡是路径中有中文字符的数据库一律提示打不开。一开始怀疑是指定了UTF8字符集的问题，因为在无法打开数据库的提示中文件名已经被转换成了UTF8编码的乱码。试着将字符集改成UTF16、UCS2、GBK、GB_2312，结果都提示不支持。**最后乾脆不指定字符集，结果反倒可以了。**

感觉问题蹊跷，研究了一下Firebird和.Net Provider的代码，终于发现了问题的起因。原来Firebird的最底层src\jrd\os\winnt.cpp line 477 PIO_open()竟然是调用CreateFileA函数打开文件，只能接受ANSI和OEM字符（还不支持UNICODE）。.Net Provider的代码更离谱，你指定什麽字符集，它就把数据库文件名转换成该字符集编码传给Firebird，在FesDatabase.cs line239 byte[] databaseBuffer = this.Charset.GetBytes(database);。这样做，在英文环境下自然没问题，在其他语言环境下就麻烦了，在Firebird的maillist中还见到过Japan人抱怨不能用Japanese路径。

解决方桉倒也简单，将.Net Provider中的FesDatabase.cs line239 byte[] databaseBuffer = this.Charset.GetBytes(database); 改为 byte[] databaseBuffer = System.Text.Encoding.Default.GetBytes(database);

这样每次打开数据库的时候.Net Provider会将文件名转换成OEM字符集传给Firebird。

作者：佚名 苏信能力支持网 http://www.jsit.edu.cn

源代码：

```
static string GetConnectionString()
 {
     FbConnectionStringBuilder cs = new FbConnectionStringBuilder();
     cs.Database = "某某工作室.fdb";
     cs.UserID = "SYSDBA";
     cs.Password = "masterkey";
     //cs.Charset = "UTF8"; //不设置任何的字符集，就可以避免出现中文路径不识别问题了
     cs.ServerType = FbServerType.Embedded; //  设置数据库类型为嵌入式
     return cs.ToString();
 }
```

2010-6-13 数据表中文字段处理成功

只需要在创建数据表的时候指定数据表的字符集编码为 gb2312 即可。

下面是测试代码：

```
private void btn_ChTest_Click(object sender, EventArgs e)
 {
     using (FbConnection conn = new FbConnection(GetConnectionString()))
     {
         conn.Open();
         using (FbCommand createTable = conn.CreateCommand())
         {
    createTable.CommandText = "create table Table_MouCH (id int, name varchar(200) character set gb2312)";
             createTable.ExecuteNonQuery();
```

```
            }
            using (FbCommand insertData = conn.CreateCommand())
            {
                insertData.CommandText = "insert into Table_MouCH values (@id, @name)";
                insertData.Parameters.Clear();
                insertData.Parameters.Add("@id", FbDbType.Integer).Value = 10;
                insertData.Parameters.Add("@name", FbDbType.VarChar, 200).Value = "某某";
                insertData.ExecuteNonQuery();
            }
            using(FbCommand selectData = conn.CreateCommand())
            {
                selectData.CommandText = "select * from Table_MouCH";
                using(FbDataReader r = selectData.ExecuteReader())
                {
                    while(r.Read())
                    {
                        //string str_Temp = r.GetString(0);
                        string str_Temp = r.GetString(1);
                        MessageBox.Show(str_Temp);
                    }
                }
            }
        }
        MessageBox.Show("OK3!");
    }
```

1 Firebird 单击嵌入式非安装版使用

Firebird 特性介绍 firebird 是一个全功能的，强大高效的，轻量级，免维护的数据库。它很容易让您从单用户，单数据库升级到企业级的应用。

一个 firebird 数据库服务器能够管理多个独立的数据库，每一个数据库同时可支持多个客户端连结。总之：它是一个开源的，强大在，可以自由使用的数据库（即使是商业上的使用）

作为一款单文件型小型数据库，Firebird 具有很多吸引人的特征，比如支持事务、支持存储过程、触发器等，而且 Embeded 版本的 Firebird 在 .NET 开发中只需要拷贝两个文件：一个 fbembed.dll（非托管但不需要注册的动态链接库）和一个 ADO.NET Data Provider 的 FirebirdSql.Data.Firebird.dll。这些特征都非常适合那些需要在客户端存储一些数据，但又不想安装数据库（比如 MSDE）软件的情形。

据称，在国外，需要使用客户端数据库的情况中，有 30%左右的开发者选择 Access，有 30%的开发者选择 MSDE 2000，有 30%的开发者选择 Embedded Firebird，剩余 10%选择其他小型数据库，如 SQLite，MySQL 等。

上面所说的 Access，MSDE 2000，Embedded Firebird，SQLite 等都是可以免费再分发（free redistributable）的数据库。相比而言，MSDE 2000 显著缺点是需要安装，最大优点是和服务器端的 SQL Server 编程模型一致，开发便利。Access 的显著缺点是功能较少，不支持事务等常用功能，最大优点是简单、多数开发者都很熟悉，部署也很方便。SQLite 支持事务，也是一款单文件数据库，比较不足的是 .NET Data Provider 还不是很成熟。Firebird 则同时具有：单文件、部署简单不需安装（只需 XCOPY 几个文件）、支持事务、存储过程、触发器，.NET Data Provider 比较稳定成熟等优点。

Firebird 本身有 SuperServer 和 Embedded 版本之分，后者只能本机访问，不接受 TCP 连接。对于开发者而言，从 Embedded 数据库切换到 SuperServer，只需更改数据库连接串中的 ServerType 值就行。

1.1 Firebird Data Provider For .NET 连接 Firebird 数据库

参考：http://blog.csdn.net/HiSpring/archive/2010/02/17/5310243.aspx

1、下载 Firebird 嵌入式数据库：Firebird-2.5.0.25920-0_Win32_embed_pdb_RC2(ZIP 格式，8.5MB) 下载解压到本地磁盘即可，无需安装。

2、利用可视化的数据库管理工具创建数据库：Firebird 的数据库管理工具有很多，本人使用了 IBManager，只有一个 EXE 文件，免安装。

3、配置数据库文件所在目录：从第 1 步中解压出来的目录下复制文件 firebird.msg 和 intl、udf 两个子目录的所有内容到放置数据库文件的目录下。注：数据库文件可以放置在任何可访问的目录上，不必和应用程序同一目录。

4、配置应用程序目录：从第 1 步中解压出来的目录下复制文件以下 6 个文件到应用程序目录：fbembed.dll、firebird.conf、ib_util.dll、icudt30.dll、icuin30.dll、icuuc30.dll。

5、引用 Firebird .NET Provider 的 dll ：先下载 Firebird.Net Provider v2.5.1 (ZIP 格式，200KB)，解压，将其中的 FirebirdSql.Data.FirebirdClient.dll 添加引用到项目。

6、连接：个人也记不住连接字符串，直接使用 FbConnectionStringBuilder 创建连接字符串，具体代码如下：

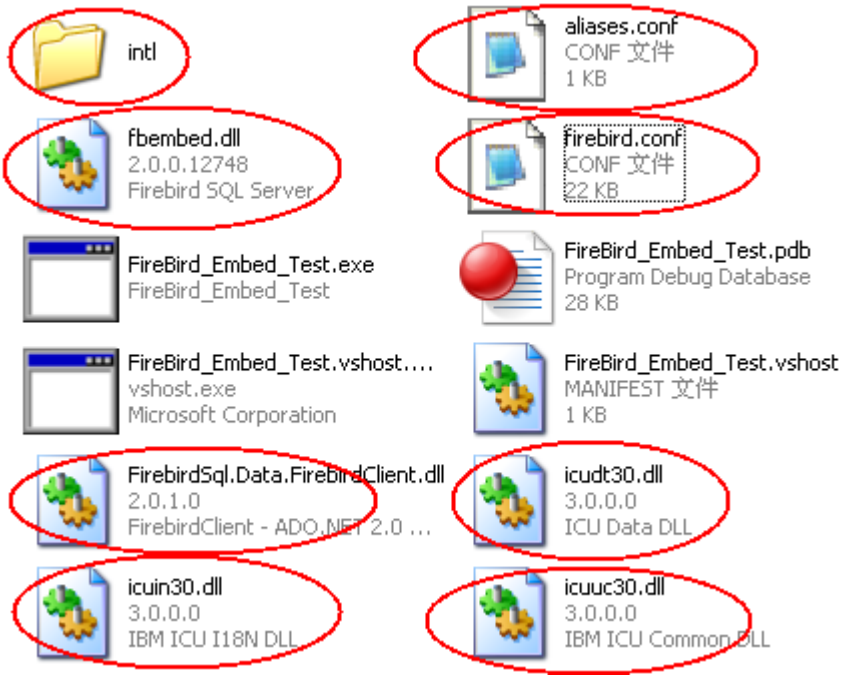
```
using FirebirdSql.Data.FirebirdClient;

FbConnectionStringBuilder connBuilder = new FbConnectionStringBuilder();
connBuilder.UserID = userId;//设置一个值，嵌入式版本并不验证用户名。
connBuilder.ServerType = FbServerType.Embedded;//设置数据库类型为 嵌入式；
connBuilder.Database = dbFile;//数据库文件的目录；

using (FbConnection fbConn = new FbConnection(connBuilder.ConnectionString))
{
    fbConn.Open();
    Console.WriteLine("连接成功！");
    fbConn.Close();
}
```

本次实验使用 2.0 版本

拷贝需要的文件至 Debug 目录：



1.2 创建 Firebird 数据库文件

```
private void btn_NewDataBase_Click(object sender, EventArgs e)
{
    FbConnection.CreateDatabase(GetConnectionString());
}
static string GetConnectionString()
{
    FbConnectionStringBuilder cs = new FbConnectionStringBuilder();
    cs.Database = "某某工作室.fdb";
    cs.UserID = "SYSDBA";
    cs.Password = "masterkey";
    //cs.Charset = "UTF8"; //不设置任何的字符集，就可以避免出现中文路径不识别问题了
    cs.ServerType = FbServerType.Embedded; // 设置数据库类型为嵌入式
    return cs.ToString();
}
```

默认的数据库用户名为 SYSDBA，默认密码为 masterkey ，但是也可以创建自己的用户名和密码，如下代码所示：

```
static string GetConnectionString2()
{
    FbConnectionStringBuilder cs = new FbConnectionStringBuilder();
    cs.Database = "某某工作室2.fdb";
    cs.UserID = "MOUSTUDIO";
    cs.Password = "851021mou";
    //cs.Charset = "UTF8"; //不设置任何的字符集，就可以避免出现中文路径不识别问题了
    cs.ServerType = FbServerType.Embedded; // 设置数据库类型为嵌入式
    return cs.ToString();
}
```

1.3 创建数据表

```
using (FbConnection conn = new FbConnection(GetConnectionString()))
{
        conn.Open();
        using (FbCommand createTable = conn.CreateCommand())
        {
createTable.CommandText = "create table Table_MouCH (id int, name varchar(20) character set gb2312)";
                createTable.ExecuteNonQuery();
         }
}
```

1.4 插入数据

```
using (FbCommand insertData = conn.CreateCommand())
                {
                    insertData.CommandText = "insert into Table_MouCH values (@id, @name)";
                    insertData.Parameters.Clear();
                    insertData.Parameters.Add("@id", FbDbType.Integer).Value = 10;
                    insertData.Parameters.Add("@name", FbDbType.VarChar, 200).Value = "某某";
                    insertData.ExecuteNonQuery();
                }
```

1.5 读取数据

```
using(FbCommand selectData = conn.CreateCommand())
                {
                    selectData.CommandText = "select * from Table_MouCH";
                    using(FbDataReader r = selectData.ExecuteReader())
                    {
                        while(r.Read())
                        {
                            //string str_Temp = r.GetString(0);
                            string str_Temp = r.GetString(1);
                            MessageBox.Show(str_Temp);
                        }
                    }
                }
```