

ADO.NET および OLE DB.NET Data Provider を使用した SQL Anywhere データベースへの接続

このマニュアルでは、Microsoft の OLE DB.NET Data Provider を使用して C# プロジェクトから SQL Anywhere データベースへの接続を作成する方法について説明します。

はじめに

Microsoft OLE DB.NET Data Provider を SQL Anywhere Native OLE DB Provider とともに使用することにより、マネージド .NET アプリケーションは SQL Anywhere データベースへの標準化されたアクセスを得ることができます。あるいは、よりパフォーマンスの高い統合ソリューションとして SQL Anywhere .NET Data Provider の使用を検討することもできます。

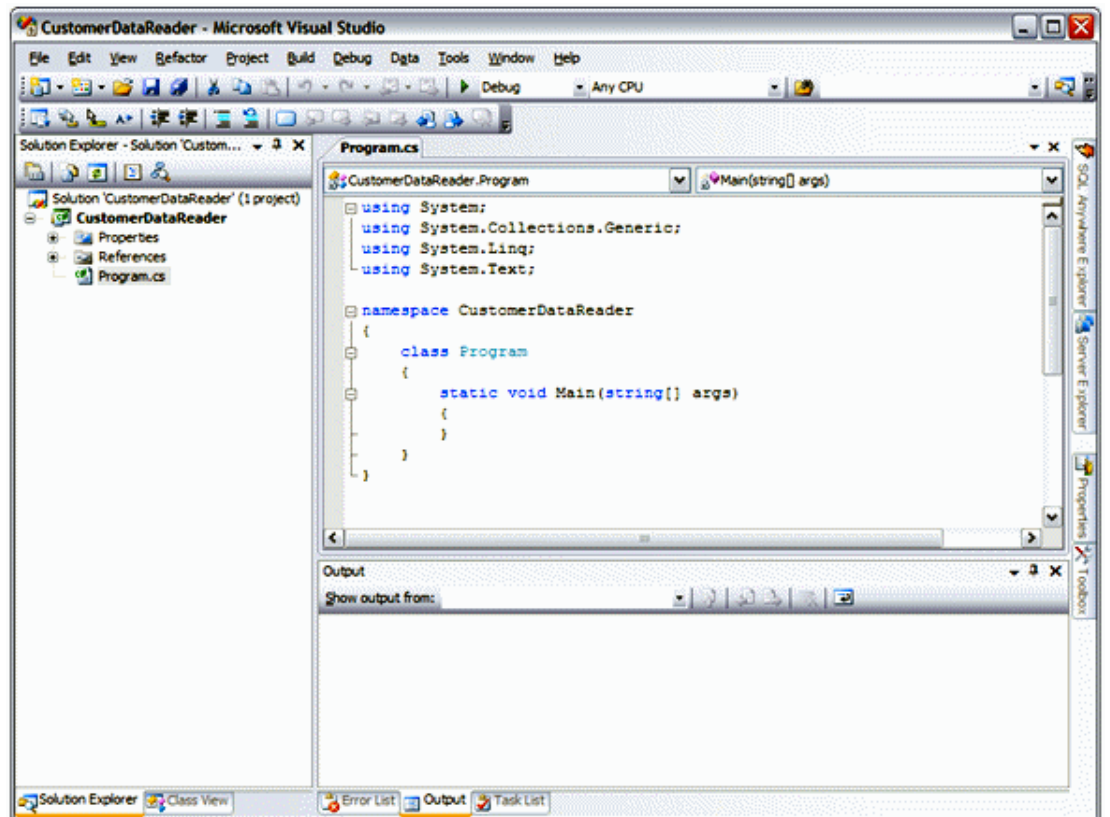
必要なソフトウェア

- SQL Anywhere 11.x
- SQL Anywhere サンプル・データベース (demo.db)
- Microsoft Visual Studio 2008

手順

1. Visual Studio 2008 を起動します。
2. 新規プロジェクトを作成します。
 - a. [ファイル] メニューから **[新規作成]** > **[プロジェクト]** を選択します。
[新しいプロジェクト] ダイアログが表示されます。
 - b. 左ペインで **[Visual C#]** を選択します。
 - c. 右ペインで **[コンソール アプリケーション]** を選択します。
 - d. **[名前]** フィールドに **CustomerDataReader** と入力します。
 - e. **[場所]** フィールドに **c:\temp** と入力します。
 - f. **[OK]** をクリックして [新しいプロジェクト] ダイアログを閉じます。

- g. しばらくすると、以下のコード・スタブを使用してプロジェクトが作成されます。



- h. 以下の using 指令をコード・ファイルの一番上にあるリストに追加します。System.Data.OleDb ネームスペースには、データベース接続に必要な ADO.NET クラスがすべて含まれています。コード・ファイルは以下ようになります。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Data.OleDb;
```

```
namespace CustomerDataReader
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

```
    }  
}
```

- i. 以下のコードを `static void Main(string[] args)` メソッドにコピーします。

```
//Create a connection  
OleDbConnection myConnection = new OleDbConnection(  
@"Data Source=SQL Anywhere 11 Demo;Provider=SAOLEDB.11");  
//open the connection  
myConnection.Open();  
//Create a command object.  
OleDbCommand myCommand = myConnection.CreateCommand();  
//Specify a query  
myCommand.CommandText = "Select GivenName, Surname FROM  
Customers"  
//Create a DataReader for the command  
OleDbDataReader myDataReader = myCommand.ExecuteReader();  
//Display the resulting Data  
while (myDataReader.Read())  
{  
    Console.WriteLine("{0} {1}",  
myDataReader["GivenName"] myDataReader["Surname"]);  
}  
  
//Close the DataReader.  
myDataReader.Close();  
//Close the connection  
myConnection.Close();  
  
//Keep the console open until a keystroke.  
Console.ReadKey(true);
```

- j. **[F5]** を押してプロジェクトを実行します。以下のような名前のリストが表示されます。

```
Michaels Devlin  
Beth Reiser  
Erin Niedringhaus  
Meghan Mason  
Laura McCarthy  
Paul Phillips
```

Kelly Colburn

....

- k. 任意のキーを押して、コンソール・ウィンドウを閉じます。

アプリケーションの仕組み

OleDbConnection オブジェクトの作成

OleDbConnection オブジェクトは、他の ADO.NET オブジェクトを使用する前に初期化する必要があります。OleDbConnection オブジェクトは、アプリケーションとデータベース・プロバイダ（この場合は SAOLEDB.11）間の接続を作成します。ユーザは残りの接続文字列を渡す必要があり、この接続文字列はデータ・ソースに含めることができます。データベース・サーバがすでに動作している場合は、ユーザ ID とパスワードを渡すだけで構いません。接続文字列は以下のようになります。

```
OleDbConnection myConnection = new  
OleDbConnection ("Provider=SAOLEDB.11;UID=DBA;PWD=sql");
```

DSN を使わずにデータベース・サーバを実行する際にアプリケーションからデータベース・サーバを起動する必要がある場合、接続文字列は以下のようになります。

```
OLEDBConnection myConnection = new  
OleDbConnection  
("Provider=SAOLEDB.11;UID=dba;PWD=sql;DBF=C:¥MyDB.db");
```

'@' 記号を接続文字列の前に置くと、ファイル名にバックスラッシュを使用することができます。そうしない場合は、C# 文字列内のバックスラッシュ文字をエスケープするために二重のバックスラッシュが必要になります。

Connection オブジェクトを開く

.NET アプリケーションとプロバイダ間の接続を開くには、このメソッドが必要です。このメソッドが失敗した場合は、例外がスローされます (System.Data.OleDb.OleDbException)
myConnection.Open();

クエリ文の指定

接続が正常に開いたら、SQL 文を発行できます。まず、データベース操作を実行するために Command オブジェクトを作成する必要があります。Command オブジェクトを作成したら、

CommandText プロパティを設定してください。顧客の特定の名前と姓をフェッチするために、その SQL 文を Command オブジェクトの CommandText プロパティに渡します。

```
OleDbCommand myCommand = myConnection.CreateCommand();
myCommand.CommandText = "SELECT GivenName, Surname FROM
Customers";
```

データの読み取り

この例では、DataReader オブジェクトを使用してクエリの結果をすぐに取得します。これは、読み取り専用のオブジェクトですので、データを更新することはできません。DataReader の Read メソッドは、ローを 1 つずつ読み取り、読み取るデータがある間は true を返し、読み取るデータがなくなったら false を返します。

```
while ( myDataReader.Read())
{
    Console.WriteLine("{0} {1}", myDataReader["GivenName"],
myDataReader["Surname"]);
}
```

クリーンアップ

最後に、DataReader オブジェクトと Connection オブジェクトを閉じます。

```
myDataReader.Close();
myConnection.Close();
```