

ADO.NET を使用して SQL Anywhere データベースに接続する

このドキュメントでは、C# プロジェクトを使用して SQL Anywhere データベースに接続する方法を説明します。

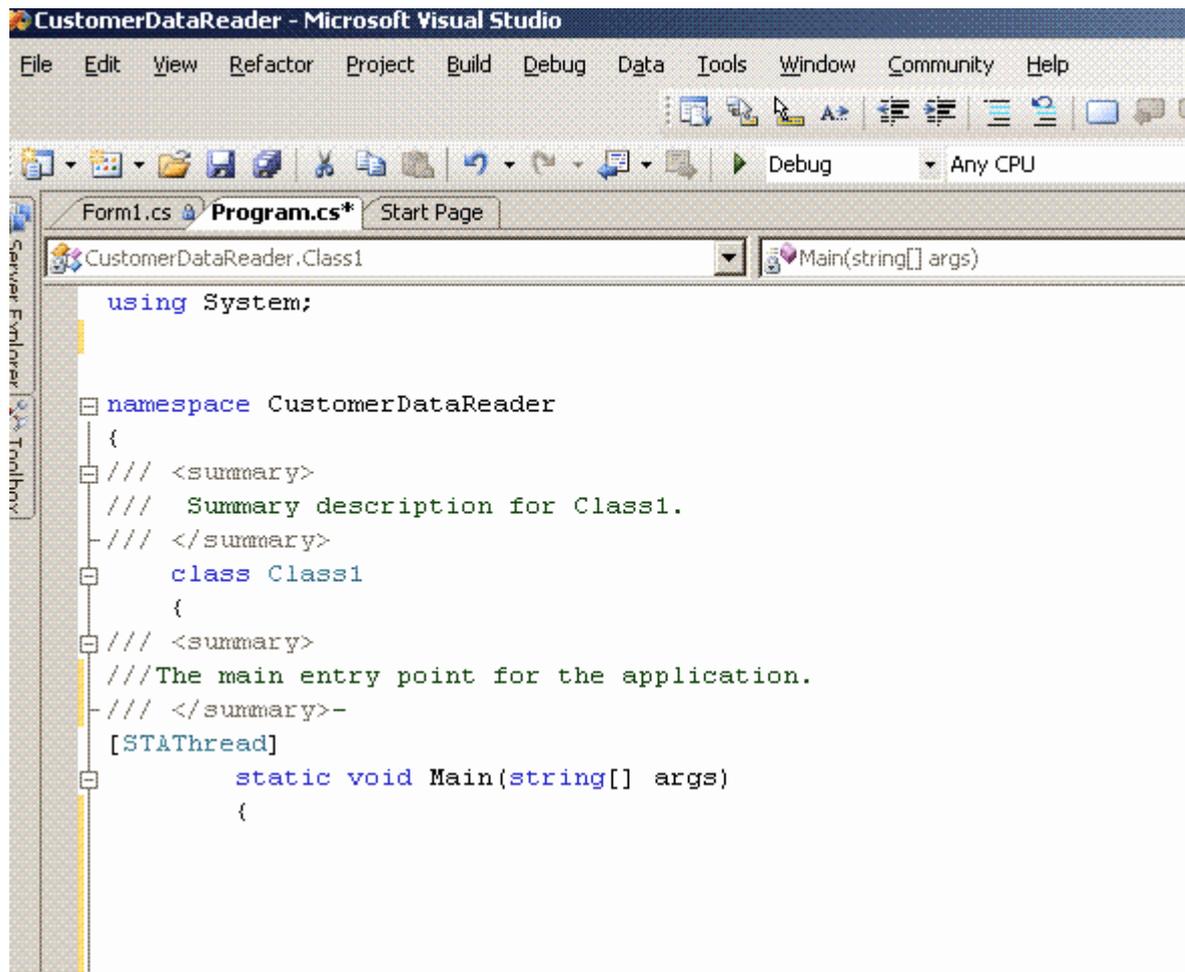
SQL Anywhere OLE DB プロバイダを使用すると、NET アプリケーションを短時間で SQL Anywhere データベースに簡単に接続できます。また、SQL Anywhere Managed Provider を使用して接続することも可能です。詳細については、「[SQL Anywhere .NET データ・プロバイダ を使用して SQL Anywhere データベースに接続する](#)」を参照してください。

必要なソフトウェア

- SQL Anywhere Studio 10.0.x
- *demo.db* ファイル (SQL Anywhere に含まれています)
- Microsoft Visual Studio .NET version 2003 または 2005
- Microsoft ADO.NET
- Windows XP/200x

手順

1. Visual Studio .NET を起動します。
2. 新規プロジェクトを作成します。
 - [File] - [New] - [Project] を選択します。[New Project] ダイアログが表示されます。
 - 左側のペインで [Visual C#] を選択します。
 - 右側のペインで [Console Application] を選択します。
 - [Name] フィールドに **CustomerDataReader** と入力します。
 - [Location] フィールドに *c:\temp* と入力します。
 - [OK] をクリックして、[New Project] ダイアログを閉じます。
以下のように表示されます。



3. System.Data の名前空間を設定するために、プロジェクトの最初に以下の using ディレクティブを追加します。これは、すべての ADO.NET クラスを配置する場所です。

```
Using System.Data;
```

4. OLE DB .NET Data Provider のプロジェクトに、以下の using ディレクティブを追加します。

```
Using System.Data.OleDb;
ソースは以下ようになります。
```

- ```
using System;
using System.Data;
using System.Data.OleDb;
```
5. CustomerDataReader の名前空間
 

```
{
```

```

 /// <summary>
 /// Summary description for Class1.
 /// </summary>
 class Class1
 {
 /// <summary>
 /// The main entry point for the application.
 /// </summary>
 [STAThread]

```

```

static void Main(string[] args)
{
 //
 // TODO: Add code to start application here
 //
}
}
}

```

6. SQL Anywhere と使用する C# アプリケーションの間の通信を確立させる以下のコードを //TODO に続く Main メソッドに追加します。

「アプリケーションをここで起動」コメントにコードを追加します。

```

//Set your connection string
OleDbConnection myConnection = new OleDbConnection(
 @"Data Source=SQL Anywhere 10 Demo;Provider=SAOLEDB.10");
//open the connection
myConnection.Open();
//Creating command object.
OleDbCommand myCommand = myConnection.CreateCommand();
//Specify query
myCommand.CommandText = "SELECT GivenName, Surname FROM Customers";

//DataReader for the command
OleDbDataReader myDataReader = myCommand.ExecuteReader();

//Display data.
while (myDataReader.Read())
{
 Console.WriteLine("{0}{1}",

myDataReader["GivenName"],myDataReader["Surname"]);

}

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

```

7. [Ctrl+F5] を押してプロジェクトを実行します。

以下のリストが表示されます。

:  
:

Michaels Devlin  
Beth Reiser  
Erin Niedringhaus  
Meghan Mason  
Laura McCarthy

Paul Phillips

:  
:  
:  
:

Jenniver Stutzman

William Thompson

## アプリケーションの動作の仕組み

1. OleDbConnection オブジェクトを使用します。

```
OleDbConnection myConnection = new OleDbConnection(
 @"Data Source=SQL Anywhere 10 Demo;Provider=SAOLEDB.10");
```

ADO.NET オブジェクトを使用するには、先に OleDbConnection オブジェクトを初期化する必要があります。このオブジェクトは、アプリケーションとデータベース・プロバイダ（ここでは、SAOLEDB.1）の間の接続を作成します。それ以外の接続文字列を引き渡す必要があります。引き渡した接続文字列は、データ・ソースに含めることができます。データベースがすでに実行されている場合は、ユーザ ID とパスワードのみを引き渡します。接続文字列は以下のようになります。

```
OleDbConnection myConnection = new OleDbConnection(
 "Provider=SAOLEDB.10;UID=DBA;PWD=sql");
```

DNS を使用していないときにアプリケーションでデータベースを開始する必要がある場合は、接続文字列は以下のようになります。

```
OleDbConnection myConnection = new OleDbConnection(
 @"Provider=SAOLEDB.10;UID=DBA;PWD=sql;DBF=c:\temp\%dbfile.db");
```

接続文字列の前に @ を記述することで、ファイル名でバックslashを使用できます。@ を記述しない場合、C# 文字列内のバックslash文字をエスケープさせるために、二重のバックslashを記述する必要があります。

2. Connection オブジェクトを開きます。

```
myConnection.Open()
```

このメソッドは、.NET アプリケーションとプロバイダの間の接続を開くために必要です。メソッドが失敗すると、例外がスローされます。

```
'System.Data.OleDb.OleDbException' occurred in system.data.dll
```

3. SQL コマンドを実行します。

```
OleDbCommand myCommand = myConnection.CreateCommand();
```

```
//Specify query
```

```
myCommand.CommandText = "SELECT GivenName, Surname FROM Customers";
```

接続を開いた後は、SQL 文を実行できます。まず、データベース操作を実行するためのコマンド・オブジェクトを作成する必要があります。コマンド・オブジェクトの作成後は、CommandText プロパティを設定する必要があります。顧客の氏名をフェッチするため、コマンド・オブジェクトの CommandText プロパティに SQL 文を設定します。

4. DataReader を使用します。

```
OleDbDataReader myDataReader = myCommand.ExecuteReader();
```

この例では、DataReader オブジェクトを使用してクエリの結果を短時間で取得します。DataReader は、読み取り専用オブジェクトです。データを更新することはできません。

以下のコードは、データを表示します。

```
while (myDataReader.Read())
{
 Console.WriteLine("{0}{1}",

myDataReader["GivenName"],myDataReader["Surname"]);

}
```

DataReader の read メソッドは、一度に 1 つのローを読み取ります。このメソッドは、読み取るデータがあるかぎり true を返します。読み取るデータがなくなると、false を返します。

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

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