Javaユーザーのための C#入門

乗り換えをもくろむあなたに

第4回 ADO.NET（データベースアクセス）

矢嶋 聡 *YAJIMA, Satoshi*
NRIラーニングネットワーク株式会社
MSDN Regional Director

はじめに

今月は、C#アプリケーションからデータベースにアクセスする方法を取り上げます。C#（厳密には.NET）とJavaのどちらもデータベースにアクセスするためのクラスライブラリが用意されています。Javaであれば、JDBC（Java Database Connectivity）が用意されており、C#ではADO.NETを利用することができます。両者は類似しているところもあり、またADO.NET固有の機能もあります。

今回は、まず類似している基本部分を解説した後、ADO.NETの大きな特徴のひとつであり、分散アプリケーション向けのプログラミングモデルとも言える非接続型の「データセット」を中心に扱います。

本稿で前提となるもの

OS	Windows 2000 Professional（SP3）以降
開発環境	Visual Studio.NET Visual C#.NET Compiler 7.00.9466 .NET Framework 1.0 Ver 1.0.3705（SP2） Internet Explorer 6.0 SQL Server 2000

初級　中級　上級

この記事で解説しているサンプルプログラムは、付録CD-ROMの¥DMAG¥CSHARPフォルダ以下に収録しています。

¥DBAPP　：本稿で解説したサンプルプロジェクト
¥DBTEST：データアクセスコード（C#版、Java版）

今回のサンプル

今回作成するC#のサンプルプログラムは、起動すると図1のような画面になります。このプログラムは、ユーザーインターフェイスとしてWindowsフォームを使っています。内部ではADO.NETを使ってSQL Server 2000にアクセスし、サンプルデータベースであるpubsデータベースのpublishersテーブル（出版社情報）とtitlesテーブル（書籍情報）の内容を表示します。フォーム上部のテキストボックスには、publishersテーブルの「出版社コード」と「名前」が表示されます。下半分の表形式の部分（データグリッド）には、titlesテーブルの中から、その出版社に該当する書籍だけを表示します。

このフォームの上部右にある［<］ボタンか［>］ボタンをクリックすると、publishersテーブル内の出版社データが1件ずつ表示され、連動して下半分のグリッドには、その出版社に対応する書籍一覧が表示されます（図2）。上

図1：publishersテーブルとtitlesテーブルの内容を表示

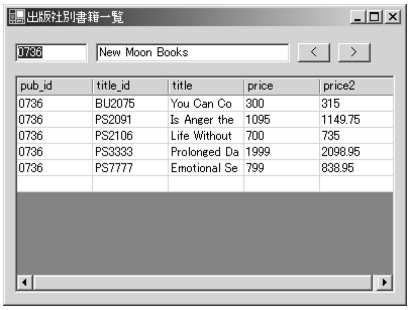
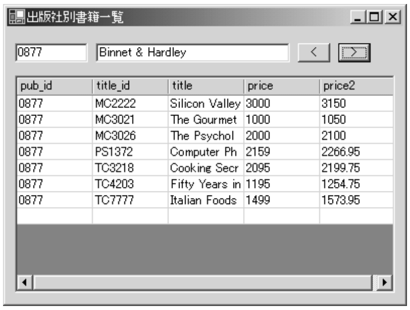


図2：上部の出版社情報に連動して、下部の書籍一覧の内容も変わる



部左の出版社コード（0877など）が、下部データグリッドの左列にある出版社コードと連動して変化する点に注意してください。

また、データグリッドの「price2」欄は、データベース上のデータではなく、price列欄の値に「1.05」を乗じた計算結果を表示しています。なお、データはすでに筆者がいろいろと加工しているので、必ずしもこのとおりには表示されないかもしれません。

今回は、プログラム内部のADO.NETを使ったデータアクセスに着目しているので、フォームのユーザーインターフェイスは単純なものにしています。しかし、このサンプルにはADO.NETの特徴的な機能が使われています。また、驚くべきことに、Visual Studio.NETで作成したこのサンプルのソースコードは、1000行を越えているのに、実際に記述したコードは、わずか7行です。

本稿ではADO.NETの特徴を説明するほかに、Visual Studio.NETを使って、これをどのように作成したかについても解説します[注1]。

ただし、このプログラムを実行するには、サンプルを実行するマシンにSQL Server 2000がインストールされている必要があります。また、このサンプルは、Windows認証を使ってSQL Serverにログインすることを前提にしています。SQL Serverを標準でインストールしてあれば、サンプルの実行者がそのマシンにAdministratorアカウントを使ってローカルにログオンし、このプログラムを実行してSQL Serverにログインできます。

ADO.NETの基礎知識

まずは、サンプルを作成する前に、ADO.NETの基本事項を確認しておきましょう。ADO.NETは、JDBCと同様に、さまざまなデータソースにアクセスできる汎用的なプログラミングインターフェイスです。JDBCの場合、各データベース固有のJDBCドライバさえあれば、データベースを問わずアクセスできるように、ADO.NETではプロバイダがあれば、データベースを問わずアクセスすることができます。

リスト1とリスト2は、今回のサンプルとは別に、基本的なデータベースのアクセスを行なっているC#版とJava版のプログラムです（実際に運用するプログラムとしては望ましくありませんが、この例では単純化する目的のため、try…catch…を使った例外処理を省き、必要な情報はハードコーディングしています）。

どちらのサンプルも、C#やJavaに限らずデータベースアクセスAPIとしてありがちなプログラミングモデルを採用していることもあって、2つのリストの構造は似ています。JDBCを知っているJavaプログラマであれば、ADO.NETにも馴染みやすいでしょう。

次に、簡単に流れを確認します。

どちらも①の部分で、データベースアクセスに必要なオブジェクトを表わす3つの変数を宣言しています。厳密に言えば、これら3つの変数は、C#ではクラス、Javaではインターフェイスであり、型の名前も違いますが、どちらも接続を表わすオブジェクト、SQL文を操作するオブジェクト、および結果セット（SELECT文の実行結果として得られるデータ）を表わすオブジェクトになっています。

②では、データベースアクセスに必要なプロバイダまたはドライバを指定して、「taro」というアカウントで接続を行なっています。C#ではOleDbCon-

注1）ここでは、ソースコード内のポイントとなるべき点だけを提示します。完全なソースコードは、付録CD-ROMに収録しているので、そちらを参照してください。また、ソースコードはVisual Studio.NETを使って作成していますが、.NET Framework SDKさえあれば、ソースコード内のForm1.csとMyDataSet.csファイルだけを、次のようにコンパイルして、実行可能プログラムを作成することができます。

```
>csc /out:DbApp.exe /t:winexe Form1.cs MyDataSet.cs
```