ADO.NETを利用したデータベースアクセス

接続型と非接続型のどちらを採用すればよいのか

株式会社CSK
教育サービス事業部
大井 卓爾
OOI, Takuji

Technology Tools

- ☑ Visual Basic .NET
- ❑ Visual C# .NET
- ❑ SQL Server 2000
- ❑ Oracle 9i
- ❑ Access 2002
- ❑ ASP.NET
- ❑ Internet Information Services
- ❑ Other:

Level

★ ★ ★ ☆ ☆

Samples

・この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥F01_02ディレクトリに収録しています。

¥MYWINAPP1
単一テーブルの1件のレコードを照会／更新

¥MYWINAPP2
単一テーブルの複数件のレコードを照会／更新

¥MYWINAPP3
複数テーブルのレコードを照会／更新

¥MYWINAPP4
複数テーブルのレコードを照会①
（2つのDataGridを使用）

¥MYWINAPP5
複数テーブルのレコードを照会②
（ひとつのDataGridを使用）

¥MYWINAPP6
複数テーブルのレコードを照会③
（JOINを使用）

はじめに

.NET環境では、Windowsアプリケーションや Webアプリケーションなど、さまざまな種類の.NETアプリケーションを開発することができます。.NETアプリケーションがデータベースにアクセスするには、ADO.NETを使用します。ただし、ADO.NETのサンプルプログラムを単に真似ただけでは、パフォーマンスや拡張性に乏しいアプリケーションになってしまう可能性があります。いきなりコーディングをするのではなく、まずはADO.NETの本質を理解する必要があります。今回は、ADO.NETの本質の理解と、いろいろなパターンのWindowsアプリケーションからADO.NETを使用するポイントを解説します。

ADO.NETの実体

ADO.NETの実体は、.NET Frameworkのクラスライブラリ内のクラス群として提供されます。ADO.NETを構成するクラス群はSystem.Data.dllに含まれています。Visual Studio .NET 2003でWindowsアプリケーションのプロジェクトを作成すると、既定でSystem.Data.dll（System.Dataアセンブリ）への参照が設定されていることを確認できます（図1）。したがって、WindowsアプリケーションからADO.NETを使用して、データベースにアクセスすることがすでに可能な状態となっています。

ADO.NETの構成を詳しく確認しましょう。ADO.NETを構成する主なクラス群は図2のようになります。データベースにアクセスするには、各クラスの機能を連携させることによって実現します。ただし、必ずしもすべてのクラスを使うというわけではありません。たとえば、データの取得に使用するクラスにXxxDataReader（以下DataReader）とDataSetの2種類がありますが、両方を同時に使うというよりは、場面に応じていずれかを選択して使用するといった具合です。今の段階では概要程度で結構ですので、まずはざっと各

図1：Windowsアプリケーションの既定の参照設定

クラスの役割を覚えておいてください。

接続型と非接続型

“接続型”は、データベースのデータ取得や更新をしている間は、データベースとの接続を持続するというアクセス方法になります。“非接続型”は、必要なときだけデータベースとの接続をするアクセス方法になります。

と、いってもなかなかイメージを浮かべづらいのではないでしょうか？そこで少し視点を変えて、実際にはどのようにデータベースからデータを取得したり更新したりするのかを、表1に示します。表を見ると、両者は単に接続方法だけでなく、データの取得と更新の方法についても異なっているのがわかります。接続型と非接続型とではプログラム自体が大きく異なるのです。

接続型

接続型の場合はデータの取得にDataReaderオブジェクトを使用します[注1]。

図2：ADO.NETを構成するクラス群

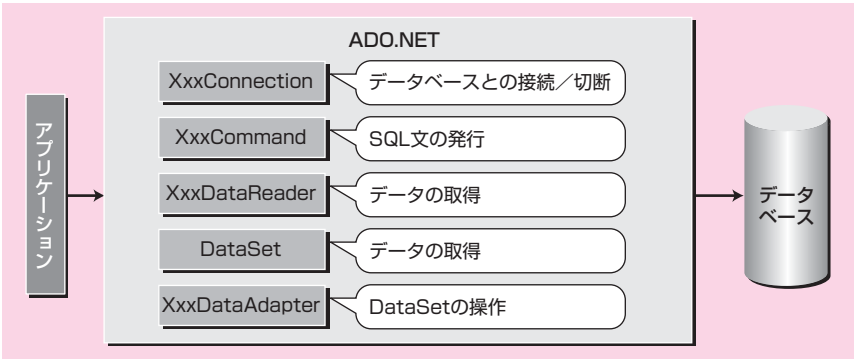

図3が接続型、つまりDataReaderオブジェクトを使用して、データベースからデータを取得する図です。

まずデータベースに接続し、データを取得するためのコマンド（SQLのSELECT文など）を実行します（①）。取得したデータはDataReaderオブジェクトに格納されます（②）。DataReaderオブジェクトに格納されたデータは、取り出して利用することができます（③）。たとえば取り出したデータを、変数に格納したり、フォーム上のテキストボックスに表示したりできます。データの取得が完了したら、DataReaderオブジェクトを閉じ、接続を切断します（④）。DataReaderオブジェクトを閉じる理由は、DataReaderオブジェクトが開いている間は、Connectionオブジェクトをほかの操作に使用できないためです。また、DataReaderオブジェクトで取得したデータは、前方向にしか参照することができず、読み取り専用であるため、直接データの編集をすることはできません。

以上を簡単にまとめると、DataReaderオブジェクトは

注1）データの取得方法によっては、DataReaderオブジェクトを使用せずにデータを取得できる場合もあります。

表1：接続型と非接続型の違い

	データ取得時	データ更新時
接続型	DataReaderを使用	任意のSQL文（INSERT、UPDATE、DELETE）を発行
非接続型	DataSetを使用	DataAdapterのUpdateメソッドによる一括更新

図3：接続型によるデータ取得