

完全掌握

DBアプリケーションなんて
カンタンだ!

SQL Server プログラミング

第

3

回

ADO.NETを使って
データベースを
操作してみよう

再入門

株式会社システムインテグレータ
湯尾 守 YUO, Mamoru
<http://www.sint.co.jp/>

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¥SQLServerディレクトリに収録しています。

¥REGISTEMP

今回作成したアプリケーションのソースファイル

¥SQL

テーブルとサンプルデータを作成するSQLスクリプトの収められたファイル

はじめに

この連載も3回目ということでしょう。よいよ今回からADO.NETを使ったデータベースアプリケーションの作成について本格的な解説に入っていきます。

ADO.NETの最大の特徴は非接続環境、つまりデータベースなどとは無縁の部分に特徴があるのですが、もちろんデータベースに接続する部分が理解できていなくては問題外であることはいうまでもありません。逆にこのデータベースに関連する部分は以前のADOと大差ない部分なのでADOなどに触れたことのある人には理解しやすいと思います。今回のテーマは、

データベースを操作するアプリケーション作成

です。

データベースアクセスの基礎

■ SqlConnectionクラス

SqlConnectionクラスはSQL Serverに接続するための機能を提供するクラスです。SqlConnectionオブジェクトのConnectionStringプロパティには接続文字列を指定します。接続文字列は「キーワード=値」の形で表記します。接続文字列の主なキーワードを表1に示します。SqlConnectionクラスにはString型の引数をとるコンストラクタもありますが、このコンストラクタを利用すると引数の値がConnectionStringプロパティに設定されます。

データベースに接続するにはOpenメソッドを利用します。そして処理が終了したらCloseメソッドを呼んで接続を明示的に閉じます。ADO.NETではトランザクションをConnection単位で管理することが

表1：接続文字列のキーワード

キーワード名	解説
Data Source	接続するSQL Serverのインスタンス名、またはネットワークアドレス
Initial Catalog	接続先のデータベース名
Integrated Security	“true” または “sspi” を指定するとWindows認証でSQL Serverにログインする。デフォルトは “false” (SQL Server認証でログイン)
Password	SQL Server認証でログインする際のパスワード
User ID	SQL Server認証でログインする際のログインユーザー名

できますが、この場合はCloseメソッドを呼ぶと保留中のトランザクションはロールバックされ、その後に接続が閉じられます。

■ SqlCommandクラス

SqlCommandクラスはSQL Serverに対してSQLコマンドを発行する機能を提供しています。SqlCommandクラスの主なプロパティを表2に示しておきます。

SqlCommandクラスのコンストラクタは、

- ・引数をとらないもの
- ・ひとつとるもの
- ・2つとるもの
- ・3つとるもの

の4種類あります。引数を3つとるコンストラクタは最初の引数の値がCommandTextプロパティに設定され、2つ目の引数はConnectionプロパティに、3つ目の引数はTransactionプロパティにそれぞれ設定されます。引数が2つ以下のコンストラクタは第3引数や第2引数などが省略されたものです。

INSERT文やUPDATE文、DELETE文を実行するにはExecuteNonQueryメソッドを呼びます。ExecuteNonQueryメソッドの戻り値はSQLの実行によって影響を受けた行数です。SELECT文を発行して値を取得したい場合はExecuteReaderメソッドを使って後述するSqlDataReaderオブジェクトを生成し、このオブジェクトを利用

表2：SqlCommandクラスの主なプロパティ

プロパティ名	解説
CommandText	実行するSQLステートメント、またはストアドプロシージャ名
CommandTimeout	コマンドの実行を開始してからエラーを生成するまでの待機時間 (秒)。デフォルトは30秒
CommandType	CommandTextに指定したコマンドの種類を表わすCommandType列挙対の値。StoredProcedure (ストアドプロシージャ)、またはText (SQLコマンド) のいずれかを指定する。デフォルトは “CommandType.Text”
Connection	コマンドを実行する際に利用するSqlConnectionオブジェクト
Parameters	コマンドが利用するSqlParameterオブジェクトのコレクション
Transaction	コマンドを実行するトランザクションを表わすSqlTransactionオブジェクト

して値を取得します。ただし、SELECT文の最初の行の最初の列のみを取得したい場合はExecuteScalarメソッドを利用することもできます。このメソッドは「SELECT COUNT(*)」など、値をひとつだけ返すようなSELECT文の場合に重宝します。

■ SqlDataReaderクラス

SqlDataReaderクラスはSELECT文の結果を取得するためのクラスです。このクラスのコンストラクタは公開されておらず、必ずSqlCommandオブジェクトのExecuteReaderメソッドを使って生成します。

SqlDataReaderクラスはSQLステートメントによって取得した列を1行ずつ読み出す機能を持っており、Readメソッドは読み込む行を1行進めるためのメソッドです。Readメソッドは1行進めたときにデータが存在していれば “True” を返し、データがなくなると “False” を返します。しかし、SqlDataReaderクラスは次の行に進める機能は持っていないも前の行に戻る機能はありません。

VB.NETのSqlDataReaderクラスにはデフォルトプロパティ (C#ではインデクサ) が設定されているので、列名または列番号を利用して配列と同じ書式で各列の値を取得できます。SqlDataReaderオブジェクトでデータを読み込んでいる間はConnectionオブジェクトがSqlData