

.NET Framework

なにを使うか、どう使えるのか アイデアノート

第7回

秋月巖ソリューション事務所
秋月 巖 AKIZUKI, Iwao
<http://www.akizuki.co.jp>

ADO.NET 自体を データベースエンジンとして使用する — その1 —

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¥NOTEディレクトリに収録しています。

¥サンプル1

DataViewオブジェクトを利用したサンプル

¥サンプル2

DataRelationオブジェクトを利用したサンプル

¥サンプル3

汎用プロシージャを利用したサンプル

ADO.NETはデータベースエンジンである

この連載の第1回（2003年11月号）で、ADO.NETはデータベースエンジンだと書いたことがある。もちろん、データベースのインターフェイスであるADO.NETを比喻としてそう表現したのではなく、「他のデータベースエンジンがなくても独立したデータベースエンジンとして使用することができる」という意味で書いたのである。テーブルを保持し、その内容にアクセスすることが可能であり、内容をファイルに入出力する機能があるという理由で最低限のデータベース機能は満たしているということができる。また、複数のテーブルを保有し、それにリレーションを設定することもできるのだから、リレーションナルデータベースエンジンだということもできる。

Xcopyライクのインストールが可能

この連載の第1回では、持ち運び可能なデータベースとして、ADO.NETの機能の一部を利用したが、うまくいけばコンプリートなデータベースエンジンとして使えるのではないかと考えた。その場合のメリットは何か。まず、あたりまえのことだが、データベースエンジンを必要としないことである。データベース操作をメインとするアプリケーションでなくても、データを内部的に管理するために簡単なデータベース機能が必要なことは多い。そのような場合、通常の実装は2つである。データベースエンジンをインストーラに組み込むか、あるいは配列をプログラムで操作するかである。しかし、メインでない機能のためにデータベースエンジンをインストールするのは気がひける場合も多いし、配列はちょっと複雑な処理に

なると操作が面倒だったりする。ADO.NETをデータベースエンジンとして使用する場合、データを操作する方法は、もちろんADO.NETと同じなので、特に新しい技術を習得する必要はない。

データベースエンジンをインストールするとなると専用のインストーラが必要になるが、ADO.NETをデータベースエンジンとして使用する場合、.NETプログラムの長所であるXcopyライクなインストールが可能になる。とはいえ、それはあくまで対象のコンピュータに.NET Frameworkがインストールされていればの話である。まあ、少なくとも.NETプログラムはインストールするコンピュータに.NET Frameworkがインストールされていることが前提なわけだから、そういう意味ではXcopyライクなインストールが可能なデータベースプログラムが開発できるといっても嘘にはならないだろう。

最大の不足点は 連結クエリーが実行できないこと

しかし、もちろん、ADO.NETはデータベースエンジンとして設計されているわけではないので、そのために使うにはいくつかの不備がある。一番、問題なのは「複数のテーブルを利用した連結クエリーが実行できない」ことである。

本来は連結クエリーが実行できなくても、複数のテーブルを個々に操作できればデータベースプログラミングをするのに問題はない。パソコンのデータベースプログラミングでSQLが使えるようになったのは割と最近のことなのである。しかし、.NETベースでのデータベースプログラミングでは、どうしても画面表示コントロールに連結して使用したい。そのためには、連結クエリーの結果を表示コントロールに連結可能な形で持つことが必要である。具体的には、DataTableオブジェクト、DataReaderオブジェクト、DataViewオブジェクトのいずれかである。

連結クエリーを実現する 汎用プロシージャを提供

そこで、今回は連結クエリーを実現する汎用プロシージャを開発して提供することにした。この汎用プロシージャを用いると、連結クエリーの結果を新しいDataTableオブジェクトとして出力する。これを併用すれば、さまざまなタイプのデータベースアプリケーションが、それほど多くの手間をかけずに開発できるはずである。もちろん、ADO.NETをデータベースとして利用した場合、すべてがインメモリで動作する。つまり、極端に大きなデータベースを操作するには不向きである。

使用するコンピュータにもよるが、今日の水準のPCならば数千件程度のデータまでならば、いい結果が得られるだろう。Microsoftのドキュメントを読むと、DataViewオブジェクトは初回の生成時にインデックスを作成するようである。つまり、パフォーマンスの問題が起こるとすれば、検索の実行時に起きるより、DataViewオブジェクトの作成時やファイルの読み込み時に集中して起きることになるだろう。特にデータのロード時間はXMLが冗長なこともあって、巨大なデータを扱う場合は問題になりやすい。あくまで、今回の試みはハイパフォーマンスを追求するというよりも、ADO.NETだけでデータベースプログラミングの機能を実装できるかどうかを検証するというのが目的である。とはいえ、パフォーマンスについては、できる範囲で考慮した。データベースプログラミングが、プログラムモジュールとしては比較的大きいデータベースエンジンから解放されるというのは十分なメリットがあると思うので、うまく用途を考えて使用してほしい。

サンプルについて

今回は3つのサンプルを用意した。

Sample 1 DataViewオブジェクトを利用することで、DataTableオブジェクトのデータから検索を実行する。