

.NET Framework

なにを使うか、どう使えるのか アイデアノート

第14回

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

SQL Serverの サーバーカーソルを使用する 自作クラス—その2—

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。

¥UDAS

UDAsクラスのコード

¥WINDOWSAPPLICATION1

前号の利用例サンプル改訂版

¥WINDOWSAPPLICATION2

UDAsクラスのPageSizeプロパティを利用して複数行をデータ連結するサンプル

複数行を連結表示する プロパティ

今回はSQL ServerのサーバーカーソルをADO.NETで簡単に利用する自作ラッパークラス「UDAsクラス」を紹介した。ADO.NETではサーバーカーソルがサポートされなくなったが、このクラスを利用すればかなり簡単にサーバーカーソルを利用することができる。

今回はUDAsクラスに変更を加え、“PageSize”プロパティを新規に追加した。このプロパティを“1”以外の値に設定すると、DataGridコントロールに複数行を連結表示することができる。ただし、この場合、データの更新はできない。また、前号のUDAsクラスはカレント行を移動するたびに、連結している表示コントロールに再連結する必要があったが、コードを見直すことでこの問題を解決した。よって、前号のサンプルアプリケーション

の改訂版(図1)も付録CD-ROM(¥WindowsApplication1フォルダ)に収録している。クラスを利用するコードがずっと洗練されたので、先月号を読んだ人はぜひ参考にしてほしい。

前号で詳しく書いたが、サーバーカーソルを使うメリットがあるのは「クエリの実行結果が大きい場合」である。その場合でも、DataReaderクラスを使えば性能は確保できるが、データ連結はできない。つまり、PageSizeプロパティの追加によって“大きい結果セットをデータ連結したい場合”にも、UDAsクラスを使用する理由ができた。ちなみに先月号で強調したメリットはDataReaderクラスと違い更新できることであった。繰り返すがデータの更新をする場合、PageSizeプロパティは“1”以外には指定できない。

さて、このUDAsクラスはDataGridコントロールに表示した場合、

図1：1行のデータを表示するサンプルアプリケーション

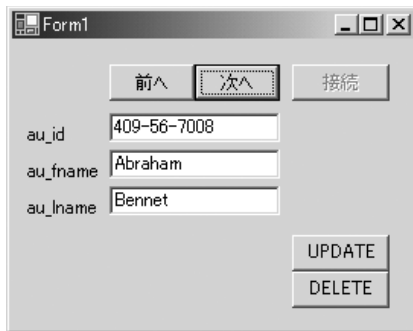


図2：PageSize プロパティによる複数行のデータ連結



ページング処理を行なうたびにSQL Serverとのラウンドトリップが発生する。最高の性能を実現するためには、一度、クライアントにロードした結果行をキャッシュしておいて、同じページが2度目以降参照される場合には再利用しラウンドトリップを防ぐことがのぞましい。DataSetクラスの構造上、そのような実装も不可能ではないのだが、次の問題をクリアするのは少々やっかいである。

図2のサンプル（リスト1）の実行画面を見てもらうとわかるが、通常、DataGridコントロールを使ったページングはコントロールのスクロールバーを使うのに対して、UDAsクラスを使う場合、この図のようにページングボタンを使用する。というのは前回も説明したように、クライアントのDataTableオブジェクトに格納されている結果セットは、カレントページのデータだけだからである。クライアントにデータをキャッシュする仕様にした場合、この切り替えをどうするかが問題になる。

複数行をデータ連結する サンプルアプリケーション

今回は新たにPageSize プロパティを利用したサンプルを紹介する。図2は [接続] ボタンをクリックし、最初のページを表示した状態である。この時点で、SQL Serverからクライアントアプリケーションが取得した結果は、ここに表示されているデータだけである。つまり、結果セットの行数が1万行あろうと、ここに表示されている4行しかネットワークを通過していないし、また、SQL Serverから読み出されていない。これがサーバーカーソ

ルを使用しているUDAsクラスの強みである。

UDAsクラスはTransact SQLのサーバーカーソルのキーセットカーソルも動的カーソルも使用することができるが、スクロール可能なカーソルを使用する必要がある。でないと、ページの後戻りができなくなってしまうからである。先に説明した、クライアントに結果セットをキャッシュする方法を採用すれば、順方向カーソルを使用することもできるようになる。ただ、以前、ADOを使って計測した結果としては、順方向カーソルだからといってスクロール可能なカーソルより速いということはありません。ただし確信はない。

いずれにしても、順方向カーソルでクライアントキャッシュを実装するならば、私だったら、Transact SQLの順方向のサーバーカーソルを使うよりもDataReaderオブジェクトを利用するだろう。とはいえ、これはあくまで好みの問題であって、DataReaderクラスのほうが.NET Frameworkの処方に忠実だからというだけの理由である。Transact SQLのサーバーカーソルを使えば、それはSQL Serverに忠実だということであり、SQL Serverと.NET Frameworkのどちらに忠実であるべきかは、意見の分かれるところだろう。

リスト1はこのサンプルプログラムの全コードである。前回、紹介したのとは違いデータの更新機能はない。それ以外はあまり変わらない。

📄 PageSize プロパティで1ページの行数を設定

このプログラムは、まず、フォームのLoad イベントプロシージャ (Form1_Load) で、次のようにPageSize プロパティを“4”に設定している。