

.NET Frameworkで作る Windows サーバー

作ればわかる アプリケーションの動作とメカニズム

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

最終回

SQLプロキシサーバーのカスタマイズ

カスタマイズの ためのプロシージャ を新たに用意

前回まではSQLプロキシサーバーをカスタマイズして、SQL Serverの代わりにJetデータベースエンジンにアクセスするJetデータベースサーバーを作成した。今回は、SQLプロキシサーバーの本来のカスタマイズ方法、つまり、SQL Serverに接続しつつ、カスタマイズによって機能を追加/変更する方法

を説明する。ここで言う機能の追加とは、SQLプロキシサーバーに新機能を追加するというより、SQL Serverにアクセスするデータベースアプリケーションの機能を追加したり変更したりする方法だと考えてほしい。以前にも書いたが、SQLプロキシサーバーを使用するメリットとして、クライアントアプリケーション、ストアードプロシージャの変更をせずにデータベースアプリケーションの機能の変更が可能である点がある。今回、紹介するカスタマイズのサンプルプログラムは、まさにそのような例である。

今回、カスタマイズをするにあたり、SQLプロキシサーバー自体にいくつかの変更を加えた。

まずひとつ目として、Jetデータベースサーバーのカスタマイズ用コードの一部をサブルーチン化し、汎用的に呼び出せるようにした。Jetデータベースサーバーは、SQL Serverから受け取ったバイナリデータのメッセージをテキスト形式のSQL文に変換する機能を有していた。この機能はSQL Serverにアクセスする場合のカスタマイズでも有効だからである。

もうひとつの大きな変更点は、SQLプロキシサーバーからSQL Serverに対してクライアントからのメッセージを転送するときに、ADO.NETを経由するか、Socketを使用して転送するかを、カスタマイズのコードで選択できるようにした。つまりメッセージ単位で経由方法を選択できるわけである。これについてはまた後で説明する

レベル >>> Level

1 2 3 4 5

言語 >>> Language

▪ Visual Basic

ツール >>> Tool

▪ Visual Studio 2005 Professional
▪ SQL Server 2000
▪ Access 2003

サンプル >>> Sample

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、<http://www.shoeisha.com/mag/windev/>からダウンロード可能です。

が、この変更の最大のメリットはカスタマイズ処理する要求はカスタマイズ用コードを通り、それ以外の場合はSQL Serverで直接処理できることである。これにより、SQLプロキシサーバーがクライアントから受信したメッセージの内容を解析し、SQLプロキシサーバーが対応しているコマンドの場合はカスタマイズでSQL文を変更し、そうでない場合はSQL Serverにそのままの形で転送するという選択ができるようになる。

サンプルプログラムの説明

サンプル1 (図1) はカスタマイズしやすいように上記の特徴を備えた新しいバージョンのSQLプロキシサーバーである。カスタマイズはしていないので、SQL Server用のクライアントアプリケーションがそのまま動作する。

サンプル2~5 (図2・4) は、サンプル1をカスタマイズしたサンプルである。これらのカスタマイズのサンプルは、次のような処理を、クライアントアプリケーション (図3) を変更せずに追加している。

サンプル2 : 特定のテーブル (table1) のデータを変更すると、別のテーブル (table2) に新しい行を追加し、変更後の内容と変更日時を記録する (ログの作成)。

サンプル3 : 特定のテーブル (table1) にデータを追加するとき、入力された内容に関わらず、同テーブルの特定列 (col_char) に、常に同じ

図3 :
すべてのサンプルで使えるクライアントアプリケーション



図1 : カスタマイズしやすいように修正した新しいSQLプロキシサーバー (サンプル1)



図2 : SQLの実行時にカスタマイズで処理を変更する (サンプル2~4)

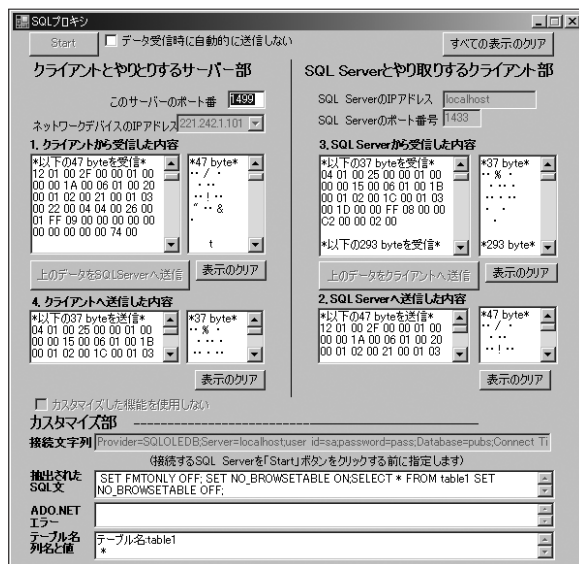


図4 : 新しいSQLプロキシサーバーでJetデータベースを再実装 (サンプル5)

