

# .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/>

第17回

## SQL Serverとクライアントプログラムを中継する SQLプロキシサーバーを作る—その2—

### カスタマイズ可能な SQLプロキシサーバー

今回はSQL Serverとクライアントプログラムを中継するSQLプロキシサーバーを作成した。SQLプロキシサーバーといっても単にTCP/IP通信を横流ししていただけ

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
- SQL Server 2005 Express
- 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.shoetisha.com/mag/windev/>からダウンロード可能です。

である。今回、用意した新しいSQLプロキシサーバー(サンプル1・図1)では次の2点の機能が追加されている。

- ①カスタマイズ用のイベントプロシーダを用意
- ②やり取りされるデータ内にあるSQL文の抽出

また、先月号ではデータの表示は「クライアントから受信した内容」と「SQL Serverから受信した内容」だけだったが、新たに、

- ・クライアントへ送信した内容
- ・SQL Serverへ送信した内容

図1 :  
サンプル1  
(カスタマイズ可能なSQLプロキシサーバー)

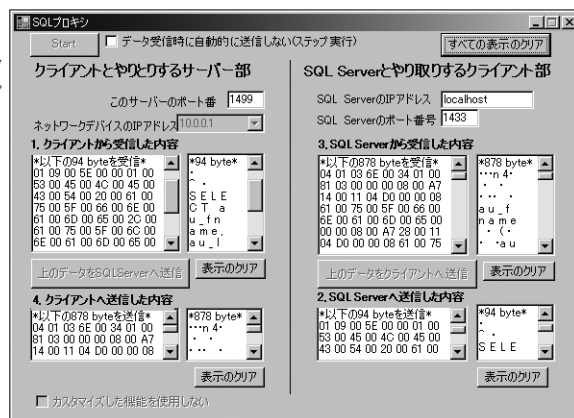
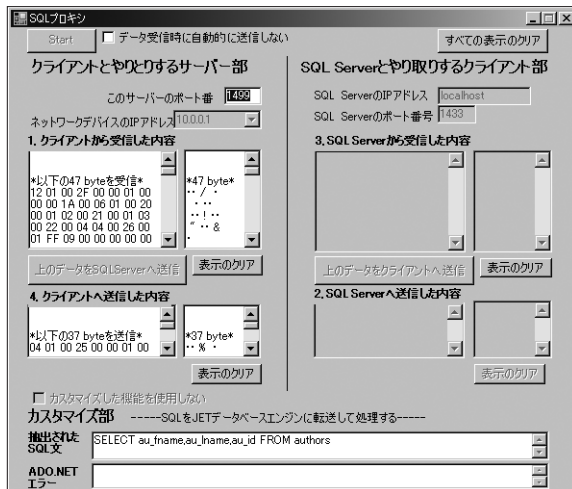


図2：サンプル2（Jetデータベースサーバー）



の2つを追加した。というのは、カスタマイズした場合、クライアントから受信したデータとSQL Serverに転送したデータの内容が異なるかもしれないからである。だから、逆にいえば、カスタマイズしていない場合、「1. クライアントから受信した内容」と「2. SQL Serverへ送信した内容」は、同じものが表示される。当然、「3. SQL Serverから受信した内容」と「4. クライアントへ送信した内容」も同じである。

SQLプロキシサーバーを使用するには、SQL Serverの設定が必要である。SQL ServerがSQL Server認証を使用し、かつ、TCP/IPで通信を行なうことを、SQLプロキシサーバーは要求するからである。SQL Serverの設定方法は記事末コラムで詳しく説明している。

なおこの後、紹介するサンプル2（図2）では、クライアントから受け取ったメッセージをSQL Serverにはまったく送信しないので、当然、返信を受け取ることもなく、右側のデータ表示部には何も表示されない。また、SQL Serverにはアクセスしないので、SQL Serverのインストールの必要もないし、もちろん何の設定の必要もない。

## SQLプロキシサーバーのカスタマイズ例

今回、カスタマイズのサンプルとして、SQL Serverに送られるメッセージをフックして、別のデータベースエンジン（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（サンプル2・図2）も用意した。

つまり、このカスタマイズされたSQLプロキシサーバーはSQL Serverなしでも動作するわけで、独立したデータベースサーバーだということもできる。SQL Serverの代わりにアクセスするデータベースエンジンには、ADO.NETを使用してアクセスしているのでOLE DBに対応したデータベースエンジンならば、なんでも使用することができる。つまり、SQL ServerのふりをしてOracleに実際の処理をやらせることも可能である。

## SQL Serverに接続するという手もある

また、サンプル2はSQL Serverに接続することもできる。SQL ServerのふりをするSQL Serverには何の意味もないように思えるかもしれないが、それは違う。たとえば、クライアントはそれぞれのユーザーIDとパスワードを持っているわけだが、SQLプロキシサーバーを経由した場合、SQL Serverからは1ユーザーがアクセスしているように見える。だから、たとえば社内データベースを利用した独自認証などを実装／設定することも可能である。

それにSQL文を実行するときには、一度、平文に変換されるので、プログラムでそれを文字列として操作することで、独自の動作にカスタマイズできる。たとえば、ある特定のテーブルに変更処理が加えられた場合、別のテーブルにも変更を加えるといったことができる。もちろん、同じことはストアドプロシージャを使っても実現することができるが、ここでポイントとなるのは、「データベースサーバーやクライアントプログラムの変更なしで実現できる」ということである。つまり、すでに完成しているプログラムに手を加えることなく、完全に後付けで処理の変更が可能になるのである。そして、その処理が必要なくなったときに、簡単に元に戻すことができる。もちろん、異種のデータベースエンジンをまたいだ処理も可能である。

ただし、まだまだ制限も多い。というのは、通信を横流しするだけの前回と違って、SQL Serverの機能のひとつひとつに対応していかないといけないからである。Web用のプロキシサーバーを思いだしてほしい。HTTPの各コマンドに対してプロキシサーバーが対応していないと、その機能は利用することができない。

