

.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

第8回

開発者が簡単に機能追加できる、
非同期通信複数スレッドTCP/IPサーバーを作る

秋月巖ソリューション事務所

秋月 巖 AKIZUKI, Iwao

<http://www.akizuki.co.jp/>



プログラマブル AMTサーバー

前回では、非同期マルチスレッド、ロードバランシング機能付きのTCP/IPサーバーを作成した。この

Level



Technology Tools

- Visual Basic
- Visual C#
- Visual C++
- SQL Server
- Oracle
- Access
- ASP.NET
- Other:

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

非同期マルチスレッドTCP/IPサーバーを「AMTサーバー」と名づけることにする。今後、この連載でサーバープログラムを作るときに、このAMTサーバーがベースになることが多いだろう。また、読者がサーバープログラムを作るときにのテンプレートにもなるはずだと考えていた。

しかし、AMTサーバーの動作は、かなり複雑である。そして前号のサンプルを元に作り直すには、AMTサーバーの動作を理解する必要がある。それでは利用しにくいのではないかと考え、プログラムをクライアントに配信するAMTサーバー部と、どのようなデータを配信するかを処理するプロシージャを分離することにした。これにより、読者は用意された空白のサブプロシージャ（「App」で始まる名称を持つイベントプロシージャ）にプログラムを記述するだけで、AMTサーバーのデータ配信機能を利用することができる。



サーバー部と 処理部を切り離す

というのは、この記事のためにSMTPサーバーを作り始めたのだが、今後、AMTサーバー部を改良したときに、それまでに作ったプロダクトすべてを修正しないといけないことに気づいたのである。でも、サーバー部のコードをまとめておいて、そこには触れないようにして開発を進めれば、AMTサーバー部をバージョンアップしたとき、それを利用したプロダクトの変更が最小限で済む。

読者の方から見れば、サーバー部がどのような処理をしているかを気にせずに、独自のサーバーアプリケーションの開発ができるということになる。つまり、コンポーネント化と同じであるが、AMTサーバーの場合、ソースコードを隠す必要がないので、ソースコードは剥き出しのままである。AMTサーバーは、あくまで学習用サンプルとして開発されており、現時点で



は、かなり不完全なので、そのほうがいいだろう。このプログラマブルAMTサーバー部に関しては、今後もソースコードを含めて一切、著作権を主張するつもりはないので自由に配布してもらってかまわない。

今回は、このプログラマブルAMTサーバーを使って開発者独自のプログラムを作る方法を説明する。だから、TCP/IPアプリケーション開発に関する説明はほとんどないので、AMTサーバーの動作について理解したい場合は前号を参照してほしい。記事末にプログラマブルAMTサーバーのリファレンスを用意したので、それを見ながら、サンプルプログラムの動作を理解してほしい。

最低限の機能拡張をしたAMTサーバー

プログラマブルAMTサーバーは、まったくプログラムがない状態では何も動作しない。通信系のコンポーネントをフォームに配置しただけでは何も動作しないのと同じである。サンプル1 (図1) はクライアント (図2) から送られたデータ内容を、そのまま送り返すプログラムである。開発者がプログラミングする必要のある全コードがリスト1である。サンプルプログラムにはAMTサーバー本体のプログラムも収められているが、リスト1以外の部分を見る必要はない。

[Start] ボタンのクリックイベントプロシージャの冒頭に書かれている次の行は、サーバーの開始に必要な値を設定している。コンポーネントでいうところのプロパティの初期値設定にあたる。

```
_PortNumber = Val(txtPort.Text)
```

この変数にはフォームの「ポート番号」テキストボックスの内容が代入される。この変数の値は、TCP/IPのポート番号として利用される。

TCP/IPサーバーを開始するには、次のように_StartServiceサブプロシージャを呼び出す。

```
_StartService()
```

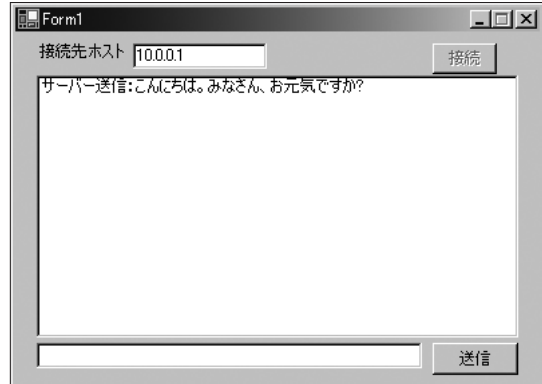
これだけでTCP/IPサービスは開始する。

これだけのコードで、クライアントはサーバーに接続し、サーバーに対してデータを送信することができる。もちろん、これだけでは、サーバーはデータを受信する

図1：プログラマブルAMTサーバーを使ってプログラミングした簡単な例 (サンプル1)



図2：サンプル1・2のためのテスト用クライアントプログラム



リスト1：プログラマブルAMTサーバーのシンプルな利用例 (サンプル1)。クライアントが送ったデータを送り返すプログラムが、これだけのプログラムコードを追加するだけで実現できる

```
'-----ここから、ユーザープログラミング-----
Private Sub btnStart_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnStart.Click
    _PortNumber = Val(txtPort.Text)
    _StartService()
    btnStart.Enabled = False
End Sub

Private Sub AsyncCallbackReceive(ByVal state As StateObject, _
    ByVal bytesRead As Integer)
    Dim InstrIndex As Integer
    Dim strSend As String
    Dim byteData As Byte()
    state.sb.Append( _
        Encoding.Default.GetString(state.buffer, 0, bytesRead))
```

```
InstrIndex = InStr(state.sb.ToString, "[messageend]")
If InstrIndex > 0 Then
    strSend = Mid(state.sb.ToString(), 1, InstrIndex - 1)
    byteData = _
        Encoding.Default.GetBytes("サーバー送信: " & strSend)
    _SendOne(state.workSocket, byteData)
    state.sb.Remove(0, state.sb.Length)
End If
state.workSocket.BeginReceive(state.buffer, 0, 1024, 0, _
    New AsyncCallback(AddressOf AsyncCallbackReceive), state)
End Sub
```

(略)
'-----ここまで、ユーザープログラミング-----