

# .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10回

## SMTPサーバーを作る-その2- ~他のSMTPサーバーにメールを転送

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



### VB.NET習得に オブジェクト指向 の知識は必要か

@ITのWebサイトを見ていたら、  
Visual Basic .NETの言語とコンパ

#### Level



#### Technology Tools

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

↓  
Visual Studio .NET 2003

#### 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.shoeisha.com/mag/windev/>  
からダウンロード可能です。

イラの設計者であるポール・ピック氏のインタビュー<sup>注1</sup>が載っていて興味深く読んだ。その中でVisual Basic .NETとオブジェクト指向について次のように言っていることが目をひいた。

不幸にも、これは過去に多くの人々が（そしてマイクロソフトさえも）「VB.NET習得のためにはオブジェクト指向を学ばなければならない」と言っていたためです。そして多くのVB6ユーザーがそのように思っています。しかし実際には先ほども述べたように、.NETプログラミングを始めるに当たってそのような知識は必要ありません。

（@IT/Insider.NET「設計者が語るVisual Basicの来た道、進む道」）

たしかに、私も「VB.NET習得のためにはオブジェクト指向の知識

は必須である」と考えていた。しかし、考えてみれば、フォームにコントロールを配置し、そのイベントプロシージャにプログラムを記述する上で、オブジェクト指向の知識はまったく必要ない。それまでのVisual Basicとプログラミングにおいて、まったく同じなのである。そして、実は、そのプログラミング方法でも、Visual Basic .NETで作成可能なすべてのプログラムが作成できるのである。

継承をサポートした本格的なオブジェクト指向であることは、Visual Basic .NETの大きなセールスポイントだった。だから、Microsoftはそれを喧伝したし、Visual Basic .NETを受け入れる人たちも、それを強く意識した。私もその一人である。

とはいってもVisual Basic .NETのプログラミングを習得するには、.NET Frameworkの知識が事実上必須であり、少なくともMicrosoftが提供するドキュメントを使って学習する上で、オブジェクト指

注1) Insider.NET「設計者が語るVisual Basicの来た道、進む道」

[http://www.atmarkit.co.jp/fdotnet/t-interview/vb2005/vb2005\\_01.html](http://www.atmarkit.co.jp/fdotnet/t-interview/vb2005/vb2005_01.html)



向の知識なしに資料を理解するのは困難だった。しかし、最近では、市販の本やインターネットのWebサイトでの解説も増えてきたので、オブジェクト指向をまったく理解せずに、Visual Basic .NETのプログラムを学習することも可能なのではないだろうか。

とはいえ、オブジェクト指向プログラミングの知識を学習すること自体が別に難しいことではないので、ついでに勉強してしまえばいいだろう。Visual Basic .NETがVisual Basicと比較して難しく感じるのは、オブジェクト指向度が高いからではなく、Microsoftが提供するドキュメントがわかりにくいからであるということを私は何度も書いてきた。もちろんそれだけが理由ではなく、.NET Frameworkの設計がわかりにくいというものもある。これも実はオブジェクト指向だからというわけではない。設計センスの問題だろう。私がdotNETマガジン2004年1月号～3月号（連載「.NET Frameworkアイデアノート」）で提供したデータベースアクセスのためのUDAクラスはADO.NETのラッパークラスだが、これはADOの書式でADO.NETを使用することができる。だから、オブジェクト指向になったことと、ADO.NETのプログラミングが複雑になったことの間には直接の関係がないのである。もっともUDAクラスはプログラミング書式を変更するためのものではなく、あくまでもDataSetクラスとDataReaderクラスのプログラミングスタイルの差を吸収するために開発したものであることを付け加えておく。

私は.NET Frameworkの設計は全般に複雑過ぎると考えているが、その機能と実装はすばらしい。.NET Frameworkをよりよく理解するためにオブジェクト指向の知識は必要かということになると、それはやはりあったほうがいい。オブジェクト指向の知識なしに.NET Frameworkのクラスを学習するとは、継承関係の把握を捨てて、すべてのクラスを独立したクラスとして解釈するということである。そうではなく、クラス間の関係性を理解していれば、つまりオブジェクト指向を理解していれば、各クラスの実装からくる固有のクセを理解しやすいし、また（理屈では）最終的な学習時間も短縮できるはずである。

とはいえ、これも何度も書いてきたが、オブジェクト指向が人間の直感的な認識システムと齟齬をきたしていることも事実だと思う。難しいことや複雑なことを克服

するのが優れた開発者なのではない。できるだけ楽な方法をとれば、問題も回避しやすくなるし、もちろんメンテナンスも楽になる。Visual Basic .NETを使って、よりコンポーネントベースライクなプログラミングを目指すというのもひとつの方法だろう。



## SMTPサーバーからSMTPサーバーへデータを送信

前回作成したSMTPサーバーは、メールクライアントとの対話ができるだけだった。これだけでは他の人にメールを送ることはできない。メールクライアントから受信したメール内容を、相手先のSMTPサーバーに接続し、送信して初めてメール送信が完了するからである。

だから今回は相手先メールサーバーに接続して送信する機能を実装する。ただし、まだMXレコードの検索は行っていないため、サンプルのSMTPサーバーでメールを送信するためには、メールアドレスの「@」以下をIPアドレスで指定する必要がある。稼動している通常のSMTPサーバーは「@」以下が自ドメイン以外の場合、メールの受信を拒否する設定のものがほとんどなので、事実上、サンプルのSMTPサーバーはまだインターネットメールの送信には使用できない。ローカルネットワークにサンプルのSMTPサーバーを2つ起動し、メールアドレスの「@」以下にIPアドレスを指定して試してほしい。相手先SMTPサーバーのIPアドレスまでもわかるならば、インターネットを介したメールのやり取りも可能である。メールアドレスの「@」前のユーザー名には何を指定してもかまわない。受信したメール内容はJetデータベースエンジンのMDBファイルに格納される。この内容をメールクライアントで読み取るのはPOPサーバーの仕事であり、それはまだ未実装である。

SMTPサーバーとSMTPクライアントがどのようなやり取りをするのかは前号で説明した。今回は、そのサーバー処理を実装したわけだが、今回は同じやり取りのクライアント部分ということになる。SMTPサーバーに接続してやり取りし、メールデータを転送する処理はメールクライアントが行なう処理と同じである。だから今回の実装部分を、メールクライアントの実装のために流用することも可能である。ただし、メールクライアントの難しいところは、文字コードの扱いなので、今回実装し