



ダイヤモンドアプリコット電話研究所
山崎 はるか
 YAMAZAKI, Haruka
<http://www.nda.co.jp/>

第3回 FTP接続に挑戦
 ~ Webカメラを作っちゃおう

level

1 2 3 **4** 5

Technology Tools

- Visual Basic
- Visual C#
- Visual C++
- SQL Server
- Oracle
- Access 2002
- 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.shoehisha.com/mag/windev/>からダウンロード可能です。

近接

流行というものが、実は「自己表現」の手段が流行していると考えたら。見るものと、作るものが“近接”していることは重要だと思う。難しいこと言ってごめんな。

パソコンなら、コンテンツを見るツールと、作るツールは、可能なかぎり同一なほうがいいという意味ね。

たとえば「ワープロ」は、本来作るツールであり、見るときは紙にプリントアウトしたものを「見る」のが基本だったはずである。実際、初期の市販ワープロのディスプレイは「数行の液晶画面」だけだった。それでレイアウトを考察するなんてことは夢のまた夢であった。しかし、PCの画面の精度および処理速度の向上によって、ディスプレイのエリアが広がると、やがてはレイアウトまでできるようになった。ワープロが「作るツール」であると同時に「見るツール」として使われるよ

うになった瞬間である。現在、もし「ワード文書」や「一太郎文書」が送られてきたら、まずは「開いて読む」のが普通である。印刷はその後の作業だろう。この行為が、これほどまでに“あたりまえ”になっているのは、すなわち、流行を通り越して“常識”の領域に到達しているからではないか？

さらに「メール」になると、「見る」と「作る」がさらに近接している。メールは、読んだら、ただちに返信できる。引用までする。SMTPクライアントとPOPクライアントとテキストエディタが分離しているメーカーなんて、まずありえない。

ところがホームページ (Webページ) になるとそうはいかない。たぶん、素人さんはこう考えるはずだ。

「ブラウザでホームページを作れないのはなぜ？」

まったく返す言葉がない。エディタで作って、FTPでアップして、ブラウ

ザで見る。こんな、ややっこしいこと、茶髪のにーちゃん、ねーちゃんができるか？ いや「見る」だけは流行ったかもしれないが、じゃあBlogブームは何か。これこそ「見る」と「作る」が近接してこそ、「自己表現」の手段となりうることの証左である。

つまりFTPなんてのは、メーラーのSMTP、POPなどのように「ツールの向こう側に隠れていなきゃいけないもの」である。

それが、どういうわけか、いまでも表に登場している。そういう代物（しろもの）だ。したがって、我々プログラマは、これを本来の姿に、すなわち、アプリケーション内部で行なわれる「プログラマ的な手段」にまで還してやらねばならん。それができてこそ一流、自分を表現できるプログラマに近づくというものである。

.NETからのFTP

FTP（エフティーピー）という言葉は、2つの意味がある。

ひとつは、「File Transfer Protocol」の略。Protocolと表現されてはいるものの、TCP/IPで動くので、実質は、

- ・キャラクター(テキスト)によるコマンド
- ・階層およびリストの出し方
- ・バイナリとテキストのアップロード／ダウンロード方法

など、ハンドシェイク部分を規定しているものと考えてよい。これを利用した「FTPサーバー」は、このプロトコルを用いてネット上でファイルを配布するためのサーバーシステムのことである。

もうひとつは、このプロトコルを実装した「ツール」という意味でのFTP。UNIXでもWindowsでも、ftpコマンドは標準で装備されている。あくまでコマンドなので、いちいちタイプして操作する。「FTPソフト」と呼ばれるのは、これらのコマンド操作をGUIベースで行なえるようにしたものを指す。Web（ホーム）ページの製作で用いられるのは、主に「FTPソフト」のほうである。

さて、FTPであろうとHTTPであろうと、Web上のコンテンツであるなら「WebRequest」「WebResponse」で取得できる。なので、これについては問題ない。今回述べるのは「アップロード」である。残念ながらVB6.0のような、FTP (Inet) コントロールはVB.NETにはない。

VB.NET環境からFTPを使う方法には、次の5つがある。

- ①Shellによるftpコマンドの呼び出し
- ②WinInet APIをDeclareで使う
- ③WebResponseをFtpWebResponseに派生させて使う
- ④TcpClientを使う
- ⑤WinSockを使う

番号を深めるごとに、低水準（下層）

においてゆくため、手間が増える。ざっくり使うなら③のWebResponse (Microsoft はマネージドとして推奨) だが、このFtpWebResponseに派生させる手段が「まことに煩雑」で、ヘタしたら、その下位層である「TcpClient」を使う方法よりもソースが増える。そこまでしてマネージドコードに付き合う義理は、我々にはない。TcpClient、WinSockについてはこの連載でもいずれ紹介するが、FtpWebResponseについては触れるつもりはないので、興味のある方は自分で調べてほしい。

ここでは①と②の方法を解説する。

ShellによるFTP

アプリケーションの保守では、ユーザーのレジストリ情報があると非常に便利ことがある。レジストリをコピーすることで、問題の再現性が高まることがよくあるし、レジストリファイルはカンタンに取得できる。たとえば、IEのレジストリだと、

```
regedit /E test.reg "HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main"
```

とすれば、test.regに再定義可能なフォーマットでファイル化される (図1・2)。こうして作った、レジストリファイルを、FTPでアップロードさせる仕組みを考えてみよう。

特定のファイルを「アップロード」