

アドバンスド

Microsoftテクノロジーを
めぐる冒険

Windows プログラミング

第 12 回

ASP.NET 2.0

こだか かおる
KODAKA, Kaoru

はじめに

最近、私の開発するアプリケーションのほとんどが、Webアプリケーションになりつつあります。当然、Visual Studio 2005がリリースされてからは、ASP.NET 2.0を利用することになりました。開発者として新しいテクノロジーに向き合うのはいつものことなので何の問題もないのですが、テクニカルライターとしては、実のところ今までASP.NETに関する記事を書くことを意

識的に避けてきました。

その理由はいくつかあるのですが、もっとも大きい理由は、ASP.NETがかなり広い範囲をカバーするテクノロジーであるということです。たとえば、ASP.NETはそれ自体が単独のフレームワークとして構成され、ADO.NETなど他のテクノロジーを使う側になります。

つねづね、Windowsプログラミングに関する連載記事の一部で取り上げるには、あまりにもボリュームがありすぎるテクノロジーだと思っていました。ともあれ、いろいろと開発経験を重ねていくと、その成果として何かしらのことを書いてみたい気になるものです。ASP.NET 2.0のすべてを説明することは到底不可能ですが、「アドバンスドWindowsプログラミング」流の切り口で、のんびりと紹介していくことにしたいと思います。

ンのASP.NET 1.0から数えて三代目になります。というよりも、.NET Frameworkのバージョンに合わせて、ということになりますけど。当初は使いづらい部分もありましたが、だいぶフレームワークとしてこなれてきたのではないのでしょうか。それまでASPで開発していた人にとっては、ポストバックの仕組みといい、ViewStateによるページデータの保存といい、大きなパラダイムシフトがありました。また、当初は開発ツールのほうも使いづらい面があり、たとえば、勝手にフォントの指定 (MS UIゴシック) がデザイン画面に挿入されてしまい、結果としてWebページのデザインが変わってしまうなどといった問題もありました。このような問題を防ぐために、せっかくのデザイナーが利用できず、常に直接HTMLを編集するという、なんだか本末転倒な状況になったりもしたようです。

そのような問題は、Visual Studio 2005で解消され、ますます使いやすく、そして開発しやすい環境が整いつつあります。また、ASP.NET 2.0には新機能

レベル >>> Level

1 2 3 4 5

言語 >>> Language

■ C#

ツール >>> Tool

- Visual Studio 2005 Professional
- SQL Server 2005 Express
- SQL Server 2000

ASP.NET 2.0の 新機能

ASP.NET 2.0はファーストバージョ

が満載で、今までいちいちプログラムを書かなくてはいけなかった処理が「コントロールを配置してプロパティを設定するだけでOK」といったケースもあります。ともあれ、細かくあげていくときりがないので、その中から「これは！」と思うものを紹介していきたいと思えます。

開発用Webサーバー

まずはじめに取り上げておきたいのが、開発用のWebサーバーです(図1)。Visual Studioとは別に、Microsoftが提供するWeb Matrixというフリーの開発ツールがありました。Web Matrixに用意されていたのが、簡易版のWebサーバーです。「Cassini」というコードネームで呼ばれていたこのサーバーツールは、すべて.NET Frameworkで作られており、ASP.NETを動作させるための機能を完全に備えています。

Visual Studio .NET 2003までは、IISを利用しないとWebアプリケーションの開発が行なえませんでした。そのため、一部のオペレーティングシステムでは開発が行なえなかったわけですが、Visual Studio 2005では、この簡易版

図2：テキストファイルへの書き込みサンプル



図1：開発用Webサーバー

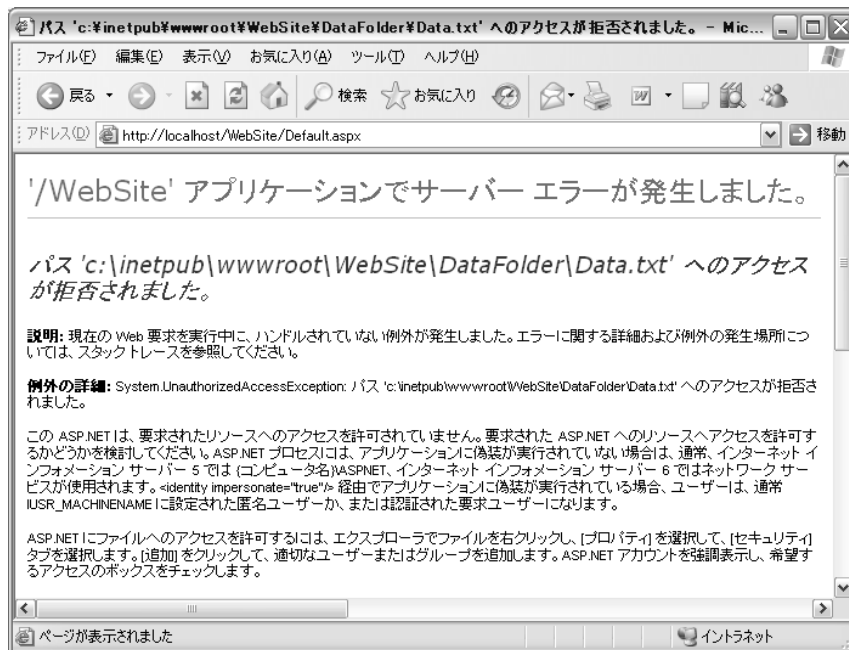


Webサーバーを利用できるため、開発時にIISが不要になりました。

これは「開発時」の利便性向上には貢献するのですが、ひとつだけ大きな問題を抱えています。それは「権限」に関する違いです。というのも、IISは相変わらずサービスとして動作していますし、開発用Webサーバーは実行ユーザーの権限で動作します。その違いが、ファイルやフォルダなどのリソースへのアクセス時に大きな違いとして現われることがあります。

たとえば、あるファイルに書き込みを行なうケースを考えてみましょう。リ

図3：IIS上で実行



リスト1：テキストファイルへの書き込み

```
protected void Button1_Click(
    object sender,
    EventArgs e
) {
    string path;
    StreamWriter writer;

    path = Server.MapPath("~/DataFolder/");
    writer = new StreamWriter(
        path + @"%Data.txt",
        false,
        Encoding.GetEncoding("Shift_Jis"));
    writer.Write(TextBox1.Text);
    writer.Close();
    writer.Dispose();
}
```

スト1では、「Data.txt」というテキストファイルにテキストボックスの内容を書き込んでいます。このサンプルプログラムを開発用Webサーバーで動作させると、何の問題も起こらずに実行できます(図2)。しかし、IIS上にWebアプリケーションを配置してから実行すると、図3のようにエラーが発生します。