

ASP.NET

ASP.NET をめぐる
思考と試行と
その冒険

I N S I D E

第13回

WebアプリケーションとWindows アプリケーションのコードを共通化する - その2 -

.NETが提供するXML Webサービスを使用しなくても
XML Webサービスを簡単に実装できる

秋月 巖 AKIZUKI, Iwao
秋月巖ソリューション事務所

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:

Level

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、付録CD-ROMの¥DOTNET¥ASP.NETディレクトリに収録しています。

¥SAMPLE
サンプルプログラムとソースコード

.NET

文字列か オブジェクトか

もう、ずい分前の話になるが、本誌にも執筆している矢澤久雄氏と「プログラミングの基本は何か?」ということについて話をしたことがある。もうひとりいた誰か(多分、編集者)が、我々に質問したのではないかと思う。矢澤氏が何と答えたのか忘れてしまったが、自分が「文字列操作」と答えたのははっきりと覚えている。文字列を自由に検索したり、切ったり貼ったりできれば、大抵のプログラムはその組み合わせで書けてしまうと思ったからである。

データベースの操作を知らなくても、配列がわからなくても、文字列さえ扱えれば、なんとかなる。しかし、この考えは、もう古いかもしれない。以前まで文字列は数値と並んでもっとも重要なデータ型であったのに、.NETの世界では膨大な数のオブジェクトのひとつにな

ってしまっているからである。

今月号でXMLを扱う上で、XML文字列を格納するのはStreamオブジェクトである。「文字列なんだから、ストリング型として扱わせてくれれば、使い慣れた方法で処理できるのに」と思ってしまう。ましてやXMLならばパーサーだって使えるのだから、データ型がストリング列であることのデメリットはほとんどない。なのにStream型の扱い方をあえて覚えなければならない。このようにオブジェクト指向プログラミングは、必要な知識が多くなる傾向がある。

.NET

文字列型か 日付型か

これまたずい分前の話になるが、知り合いの非常に優秀なプログラマが「自分はデータベースプログラミングをするときに日付型を使わない」と私に話したことがある。私はデータベースで日付を扱うとき

に、日付型を使うことに疑いを抱いていなかったの、かなり驚いた。全部、文字列型として処理するのだそうである。それを聞いた瞬間、それでは日付型のもつ機能が使えないから、さぞかし不便だろうと思った。データベースに日付データを格納するときは、彼は文字列として格納し、操作するときには自作した簡単な関数を使うという。

また、別の機会に会ったやはり優秀な開発者も同様の方法を使うという。もしかしたら、汎用機の世界では当たり前の手法なのかもしれない。私はこの方法の問題点を考えた。もし、それでことが済んでしまうのならば、日付型などというものをデータベースの設計者が用意しないはずである。しかし、考えるほどに、実はこの方法で問題がないことに気づくことになる。そして高級なデータ型がもつ固有のトラブルに巻き込まれる可能性は、ずっと低くなる。

.NET 単純なもの複雑なもの

この2つの話の本質は、同じである。どちらも、簡単な文字列で済むことを、より高級で複雑なオブジェクト(型)を使用して操作させている。このような機能を提供する側の言い分は「覚えれば便利」である。この言い分の問題点は次のようなことだと言えるだろう。

覚えないと不便

きちんと動かなかったり、動かし方がわからなかったりするので、覚えても不便なことがある

覚えれば便利だが、従来の方法でも最初に簡単なライブラリを書けば同じくらい便利

については.NETの場合、クラスライブラリのヘルプの説明が舌足らずなので、さらに不便である。ただ、いずれにしても提供されるライブラリの質が最重要であることは間違いない。今回使ってみて、やはり、.NETはXMLにかなり力をいれて作られていると感じた。

こういうことを書いてみると、じゃあ、今まで通りに文

字列を使ってプログラミングすればいいじゃないかと言われそうである。しかし、そうもゆかないのである。.NETではメソッドがオーバーロードされているため、「引数に文字列を指定する場合」と「Streamオブジェクトを指定する場合」ではふるまいが違ってしまふ。

たとえば、今月号ではADO.NETのReadXmlメソッドを利用しているが、このメソッドの引数にStreamオブジェクトを指定した場合は「XMLデータ」として扱ってくれる。しかし、文字列を指定した場合は「XMLデータが格納されたファイルのパスを示す文字列」と判断してしまうのである。

.NET XML Web サービスを使わなくても XML Web サービスを実装できる

今月はVisual Basic .NETを使ったWebアプリケーションやWindowsアプリケーションから、他のASP.NETプログラムを呼び出して使用方法を解説する。今回は.NETのWebClientクラスがaspxファイルをWebサーバー経由で呼び出す方法を採用するが、同じ処理をXML Web サービスを使用して実現することもできる。実際、次号では同じことをXML Web サービスを使って実装する。

WebClientクラスとaspxファイルを使用する方法の問題点は、WebClientクラスがあくまでもWebブラウザのふるまいをシミュレートするクラスのため、データのアップロード機能が貧弱だということである。XML Web サービスならば、この点でさらに柔軟性があるだろう。とはいえWebアプリケーションとプログラムを共有する限り、アップロードするデータはWebブラウザの入力機能の制限を受けてしまうのだから、それに準じた機能しかないということは致命的な差とは言えないかもしれない。

また、両者のコードの記述にどのような差が出るのかも、実際のプログラミングを通して比較してゆきたい。どちらの方法がいいのか、私自身まだ結論はもっていない。もし、最終的な機能に差がないならば、XML Web サービスではなく、WebClientクラスとaspxファイルを