



# Object BLVD オブジェクトの散歩道

例題でわかる！ .NET Framework

## 第 6 回 「XML のさわり」

吉田 弘一郎 YOSHIDA, Koichiro

### 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¥SAMPOディレクトリに収録しています。

¥SAMPO06¥SAMPO06.SLN  
今回紹介したサンプルのソリューションファイル

¥ANSWER  
練習問題の解答



### XML の不思議

考えて見ればXMLほど不思議なものはありません。解説書を買うのもためらうほど簡単なものにも見えれば、その裏には恐ろしく複雑な世界が隠されています。アルゴリズムを重視するソフトウェア開発者には理解しがたいドキュメントの世界なのです。一見簡単で、その反面何種類もの「法体系」のようなものが存在し、まるで企画倒れのJavaのような世界です。

ところが、おかげさな規則に見合うほどの見返りがあるようにも思えます。それなのに、お互いに敵対関係にあるはずの大手ソフトメーカーが、ことXMLに関しては揃って積極的な姿勢を示しているという摩訶不思議な代物です。幸か不幸か、ソフトウェア開発者にとってXMLというのは避けて通れない存在になってしまいました。

Javaの場合には、最初はXMLなんて視野にありませんでしたが、気が付くと非常に積極的にサポートしています。これも、Java本来の非常に排他的な

Java原理主義とは相容れないものであるはずなのですが気にならない様子。Javaとは対照的に後発の.NETの場合には、XMLは最初から堂々と取り入れられています。実際、データベースやWeb関係で.NETを用いると、気が付かないうちに結構XMLを使うことになるのはみなさんもお存知のことと思います。Visual Studio .NETに話を限っても、「ファイル名拡張子がXMLではないが、中身がXML」であるファイルが何種類もあります。ですから、この連載で、XMLのさわりの部分を話題にしても、みなさんの多くには“釈迦に説法”かもしれません。でも、XMLをゼロからプログラムするという機会も、自分で作らなければ巡ってこないと思うので、今回はあえて初歩的なXMLを話題にします。

### 例題 1 : 簡単なXML文書の作成 (VB01.vb)

UMLのダイアグラムを描くために専用のソフトが必要か否かは、大いに意見の分かれるところです。売り手の論



理にさせられて「UML = 自動コード生成」と思い込むと、相当値の張るUMLツールを買う羽目になるようです。Javaプログラム作成にも、HTML文書作成にも似たような側面があります。

もちろん、グラフィックスが絡む部分は「グラフィカルに扱うのが便利」であることは言うまでもありませんが、HTMLの兄貴分であるXML文書の場合には、なぜか通常ほとんどグラフィックスは関係してきません。ですから、XML文書の作成は、高価なツールを使う必要なんてまったくなさそうです。とくに、最初の例題で扱う図1のように簡単なXMLの場合には、NotePadでも十分でしょう。

図1の「SID」とはStaff IDのつもりですが、このように簡単なXML文書を、ここではC#やVisual Basic .NETで作ったプログラムを用いて作ろうというのです。その理由は簡単、「XMLを読み込むための準備体操」なのです。一般的な話として、次のような事実に注目してください。

XMLを作るのは楽でも読み込むのは面倒。

これは、何もXMLに限った話ではありませんが、とりわけモダンなオブジェクト指向言語を使い始めることになると、テキストファイルを読み込んで処理する能力が極端に低下するのです。ですから、XML文書を読み込んで、そこから必要な情報を抽出するのは相当面倒な作業になることを覚悟するべきなのです。C#やVisual Basic .NETも、

その例外ではありません。ただし、ここでは「面倒」であると言っているのではなく、決して「困難」であるわけではありません。ただ、原理的に簡単なことをするわりには、実際の作業が冗長になるのです。ですから、次のように言っておきましょう。

XMLの読み込みには既成のライブラリを活用すべし！

すなわちXML用ライブラリの選択が非常に重要になってきます。実際、Java、C/C++、Perl.....と何種類もあるプログラム言語ですが、これらで利用できるXML用ライブラリにも多くの種類があり、その選択には頭を悩ませることになりそうです。

しかし、.NETの場合には、XML用ライブラリはフレームワークに含まれているので迷うことはありません。幸いなことに、.NETのXML用ライブラリ（正確にはクラス）の評判は上々で、安心して使えるような雰囲気。

このように、.NETでは、XML文書の読み込みには.NETの一部になっているXML用クラスを活用することになるのですが、これらはもちろん読み書きの両方をサポートします。どうせ読み込みで使うのですから、最初は書くのに用いてみましょう。

XML文書の作成には何種類かの方法がありますが、ここでは何も考えずに、普通にプリントする要領でXML文書を書き出すためのクラス「XmlTextWriter」を用います。“牛刀で鶏をさばく”感じになりますが、これもXmlTextReaderクラスを用いてXML文書を読み込むた

図1：例題1で作成する簡単なXML (VB01output.xml)

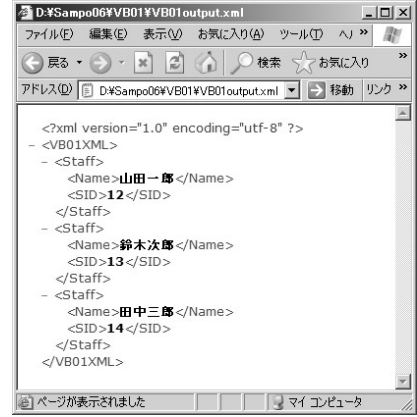


図2：例題1の実行結果



めの準備体操とってください。

注意するのは、Visual Studio .NETでC#やVisual Basic .NETを用いる場合、出力ファイルの出力先ディレクトリはEXEのディレクトリになるということ。C++の場合は、そのプロジェクトのソースファイルの入るディレクトリなのですけどね。

早速、例題1のソース(リスト1)をご覧ください。非常に単純なXMLファイルを、文字通り馬鹿丁寧に作っています。実際、テキストエディタで作るほうがはるかに早いと思いますが、ここでは「XML文書の構成部品の名前を覚える」ためとってください。「ここでエレメントが始まって、その中でまたエレメントが始まって.....」と、ソースを見ながらXML用語を追ってください。

実行結果は、図2の通り。コンソールには何も出力されない代わりに、ソー