



# Object BLVD オブジェクトの散歩道

例題でわかる！ .NET Framework

## 第 7 回 「XMLでシリアル化」

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:  
XML

### 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。

¥SAMPO07¥SAMPO07.SLN  
今回紹介したサンプルのソリューションファイル



先月に続いてXMLの話です。先月は、正面からXMLに取り組んだので少々疲れましたね。今回は、XMLそのものには直接手を触れず、上手に応用することに徹しましょう。実際、XMLを自分で直接扱う機会は極めて少ないものと思われます。ちょっとしたXML文書を書き出す程度ならまだしも、XML文書の読み込みルーチンを自らプログラムする羽目に追い込まれることは、ほとんどないはずだし、逆にあってはならないのです。“気が付いたらXMLを使っていた”という状況のほうがはるかに多いのです。

今月のテーマは「XMLを用いたシリアル化」です。すると、みなさんは即座にASPの話かなと思われるかもしれませんが、ましてや、SOAPも使うということになればなおさらです。実際、ASPを使えば、あちこちにXMLが出てきます。しかし、ここでは、ASPなどよりはるかに初歩的なレベルで、XMLを用いたシリアル化の例題をこなしてゆく

ことにします。要するに、XML文書に情報を書き出したり、逆に読み込んだりするのですが、ここではXMLの細かい部分にはまったく触れることなく、一種の馬鹿チョンモードで知らないうちにXMLすることにしましょう。



シリアル化 (Serialization) とは、文字通りシリアルにすること、つまり、構造をもっているかもしれないデータを、さらにはオブジェクトを、バイトのストリームにしてしまうことです。ここでは、XMLという条件を付けますので、バイトのストリームとは、実はXML文書となる文字列の流れになります。要するに、データとかオブジェクトをXML文書に変換してしまえば、ファイルなどに格納したり、ネットワークで遠方に転送したりということが可能になるという話です。

それでは、何をシリアル化しましょう。「データとかオブジェクト」では、漠然としすぎていますので、ここ

では「オブジェクトのシリアル化」に話を限定してしまいます。しかし、まだ一般的過ぎます。オブジェクトは単なるデータではないからです。

「データ構造+アルゴリズム=オブジェクト」ですから、オブジェクトのメソッドまでもシリアル化してしまうのでしょうか？ もちろん、これは可能です。そもそも、PCで実行可能なプログラムは、たとえばABC.EXEのような実行形式のファイルに収まっていますが、これは「バイナリストリームにシリアル化されたプログラム」であり、これがローダーによってメモリ上に逆シリアル化され、プログラム本体が再現するのです。しかし、ここでは、そこまでは望みません。それどころか、次のようなレベルにまでシリアル化を限定してしまいます。

オブジェクトの外部から読み書き可能なパブリックなデータメンバのみの自動化されたXMLシリアル化

たとえば、次のようなStaffクラスがあるとします。ここには、3つのパブリックなデータがあり、これ以外にはパブリックでないデータやさまざまなメソッドがあるとします。

```
Public Class Staff
    Public SID As Integer
    Public Name As String
    Public EAddress As String
    ...
End Class
```

今月取り扱うXMLシリアル化の場合には、このStaffクラスのオブジェクトの3つのパブリックなデータのみを対象

にするのです。

## 例題1：単一のオブジェクト(VB01.vb)

例題1のソースコードVB01.vbをご覧ください(リスト1)。Visual Basic 6.0では、クラス定義はクラスファイル、プログラムモジュールはプログラムファイルという具合にファイルが単位になっていましたね。しかし、Visual Studio .NETでは、ひとつのソースファイルの中にクラス定義とプログラムモジュールを混在させることができます。この例題1でも、クラス定義とプログラムの部分を行き来しながら、どんな例題にしようかなど考えることから始めたのですが、ひとつのファイルで済むというのは便利です。

さて、この例題の核心は、XMLシリアライザー“XmlSerializer”です。これを用いると、オブジェクトが一発でXML文書に変換されてしまうのです。XmlSerializer型オブジェクトをserializerとすれば、Staff型オブジェクトのTomは次の1行で出力先のファイルwriterにXML文書として書き出されてしまうのです。

```
serializer.Serialize(writer, Tom)
```

この1行でできたXML文書をInternet Explorerで見ると図1のとおり。たしかに動作しています。先月は、このようなXML文書を手作業で組み立てていましたが、ここでは一発でできてしまいます。たしかに便利です。

XMLシリアライザーは次のようにして作ります。生成子のパラメータは「シリアル化したいクラス」です。

```
Dim serializer As XmlSerializer = _
    New XmlSerializer(GetType(Staff))
```

出力用ファイルは、次のようにストリームライターを作っておけばよいでしょう。

```
Dim writer As StreamWriter = _
    New StreamWriter("../Vb01.xml")
```

こうしてできたXMLシリアライザーのSerialize関数を用いて、オブジェクトを出力ファイルにXML文書として書き出せばよいのです。

```
serializer.Serialize(writer, Tom)
```

また、XMLシリアル化の対象となる

図1：例題1で書き出したXML文書 (Vb01.xml)

