

将来にわたって使えるスキルを身につけるために

オブジェクト指向で はじめる プログラミング

日向 俊二
HYUGA, Shunji

第 8 回

C#でXML文書を扱う (その2)

Technology Tools

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

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¥C#SHARPディレクトリに収録しています。

¥XMLGLOSS
今回のサンプルプログラム

※この記事では.NET Framework SDKを利用して解説していますが、C#Builder、SharpDevelopでも利用できます (C#コンパイラならすべて利用できます)。



はじめに



今回は、XML文書を扱う単純なプログラムをC#で作成してみました。C#のプログラムとXML文書は、“オブジェクト”というキーワードで深く関連しています。

XML文書を扱うためのアプリケーションインターフェイスとして、W3Cで標準化されている「DOM」があります。このDOMを使うと、C#のプログラムとXML文書がオブジェクトというキーワードで関連していることがさらに明確になります。そこで、今回はDOMを使うC#のプログラミングについて見てゆきましょう。



DOMとは?



DOM (Document Object Model : ドキュメントオブジェクトモデル) は、プログラムやスクリプトなどからXML文書 (XMLドキュメント) の構造にア

クセスして、参照や更新を行なうためのプログラミングインターフェイスの仕様です*1。DOMはプラットフォームやプログラミング言語に直接関係しないように設計されています。そのため、C#を含むさまざまな言語から利用できます。また、XML文書だけでなく、XMLで整形形式文書と呼ばれる形式で記述されたHTML文書やXHTML文書も扱うことができます*2。

DOMの目的は、XMLやHTMLで書かれたドキュメントの内容や構造を調べたり変更したりすることです。そのため、DOMは「ツリー構造」というXML文書の特徴を利用しています。

DOMでは、XML文書の内容全体をツリーの形でそのままメモリ上に保存し、定義されているインターフェイスを使ってその各要素にアクセスします。つまり、XML文書のツリー構造を保ったままメモリ上にそっくり読み込んで、その全体や内容の一部を取り出して、表示したり加工して出力でき、さらにはツリーに枝葉を加えたり、まったく新たなXML文書のツリーを作成するこ

- ★1 現時点でDOMの仕様にはLevel 1、Level 2、Level 3があります。Level 2とLevel 3はLevel 1をベースに拡張された仕様です。本稿の内容はLevel 1の範囲内であり、名前空間やXML Schemaなどを使わない単純なXML文書はLevel 1の範囲内で扱うことができます。
- ★2 これに関しての詳細を知りたい場合は、記事末の参考リソース①を参照してください。

とさえできます。

◆DOMによるXML文書の操作

DOMがXML文書を扱う手順を、さらに詳しく見てみましょう。

- ①XML文書を構成する要素をそのままメモリ上にツリー形式で保存する
- ②ツリーの各要素や特定の要素より下の一連の子要素をそれぞれ“オブジェクト”として扱う

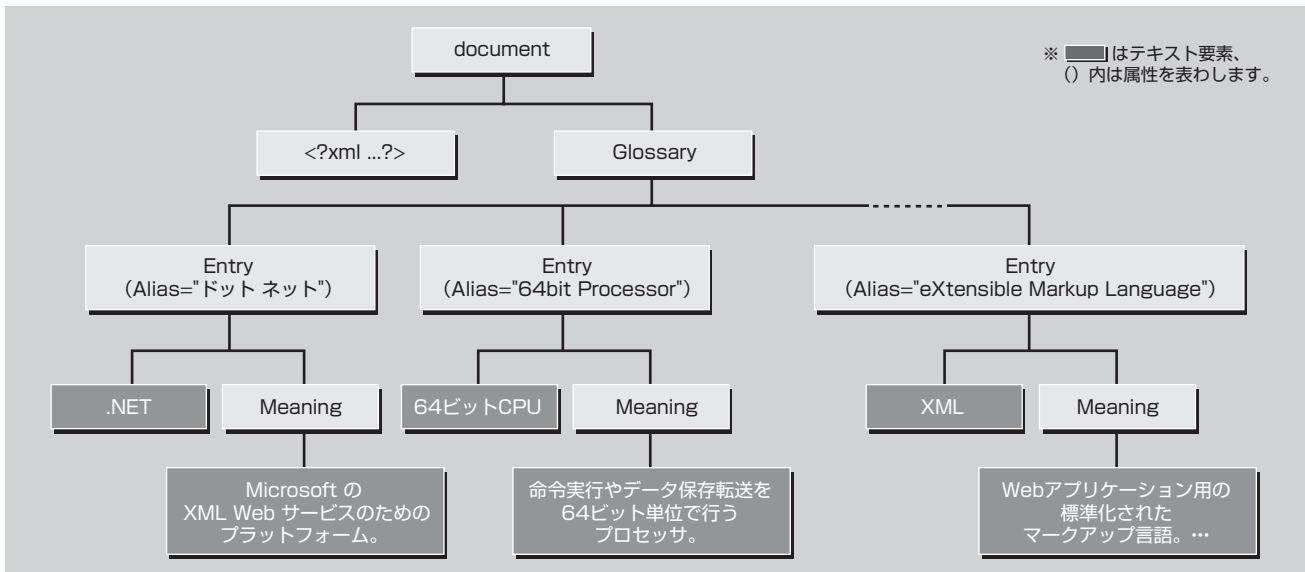
ここで“要素”と呼ぶものは、DOMの用語で正確にいえば「ノード」であり、XML文書の要素とは少し違います。たとえば、リスト1のXML文書の構造をDOMの流儀で図式化すると、図1のようになります。リスト1をInternet Explorerで表示したときの状態は図2のようになり、これも合わせて見ると理解しやすいでしょう。

さて、この図1を見るとわかるように、「<タグ>テキスト</タグ>」のような“XML文書の要素”は、「タグ」という

ノードと、その子ノードである「テキスト」ノードに分けて考えます。また、文書 (document) 全体もノードですし、XML宣言もひとつのノードです。

DOMではあらゆる独立可能な構成要素をノードと考えます。タグに挟まれたテキストはもちろん、文書全体からコメント、そして状況によってはひとつの改行や、(特に無視することを指定しない限り) 有効なスペース、ドキュメント全体もひとつのノードとみなします。

図1：リスト1のXML文書の構造



リスト1：XML文書のサンプル

```
<?xml version="1.0" encoding="UTF-16"?>
<Glossary>
  <Entry Alias="ドット ネット">.NET<Meaning>Microsoft の XML Web サービスのためのプラットフォーム。</Meaning></Entry>
  <Entry Alias="64 bit Processor">64ビットCPU<Meaning>命令実行やデータ保存転送を64ビット単位で行うプロセッサ。</Meaning></Entry>
  <Entry Alias="「エイリアス」または「エイリアス」と読みます。">alias<Meaning>別名のこと。Aliasクラスは別名を付けるためのクラスで、VB.NETのDeclare ステートメントで使われるキーワードAliasはプロシージャに別名を付けます。shやcshなどのシェルで別名を定義するときを使うUNIX系OSのコマンド。</Meaning></Entry>
  (略)
  <Entry Alias="XML Linking Language (XMLリンク言語)。">XLink<Meaning>XML Linking Language (XMLリンク言語) の通称で、XML文書の任意の位置にリンクできるようにするために策定された仕様。</Meaning></Entry>
  <Entry Alias="eXtensible Markup Language">XML<Meaning>Webアプリケーション用の標準化されたマークアップ言語。現在では、Web上のあらゆるドキュメントの事実上の標準フォーマットとして使われるようになってきました。典型的なXML文書は、&lt;?xml version="1.0"&gt;というXML宣言で始まり、以降にXMLインスタンス (XML文書の内容) が続きます。</Meaning></Entry>
</Glossary>
```