

C# and .NET Framework

Bài 7: XML in .NET

Đoàn Quang Minh

minhdqtt@gmail.com

<http://www.VTPortal.net>

Last update: 30. December 2006

Mục lục

- XML Overview
- Đọc XML
- Ghi XML
- DOM trong .NET
- Ứng dụng

XML Overview

■ XML là gì

- XML – Extensible Markup Language: ngôn ngữ định dạng mở rộng.
- XML được định nghĩa bởi w3c (World Wide Web Consortium): tổ chức mạng toàn cầu.
- Thực chất, XML là ngôn ngữ tổng quát dùng định nghĩa dữ liệu thông qua các thẻ.

■ Ngôn ngữ định dạng

- Bao gồm một tập các thẻ, và dữ liệu chứa trong các thẻ đó. Ví dụ, HTML là một loại ngôn ngữ định dạng.
- Trong XML, các thẻ không hạn chế như HTML. Người sử dụng có thể tự do định nghĩa các thẻ của mình.
- Các ưu điểm:
 - Dễ dàng trao đổi dữ liệu: do khuôn dạng XML rất dễ hiểu. XML còn được gọi là dữ liệu tự mô tả.
 - Khả năng tùy biến cao: việc cụ thể hoá các thẻ của XML sẽ tạo ra một loạt các ngôn ngữ mới, ví dụ MML, CML.
 - Có thể lưu dữ liệu có cấu trúc: dựa trên việc sử dụng các thẻ lồng nhau.

XML Overview

```
<? xml version="1.0" encoding="UTF-8" ?>  
<Document>  
  <Greeting>  
    Hello from XML  
  </Greeting>  
  <Message>  
    Welcome to the wild and woolly world of  
    XML.  
  </Message>  
</Document>
```

XML Overview

■ Các ứng dụng cụ thể của XML

- XHTML: đây là mở rộng của HTML. Về bản chất, XHTML dùng các thẻ của HTML, các thẻ này phải viết thường, và tài liệu XHTML phải là tài liệu hợp khuôn dạng.
- CML, MML: các ngôn ngữ biểu diễn công thức hoá học, công thức toán học. Các công thức này có thể được vẽ trên các trình duyệt đặc biệt, thông qua dữ liệu XML.
- WML: ngôn ngữ định dạng mạng không dây, nhằm tạo các tài liệu web hiển thị trên máy điện thoại di động.
- SOAP (Simple Object Access Protocol): giao thức truy cập đối tượng đơn giản, cho phép các ứng dụng web có thể trao đổi thông tin với nhau. Khi một ứng dụng web cần thi hành một chức năng do một web service cung cấp, nó sẽ gửi dữ liệu theo chuẩn SOAP tới web service. Đến lượt mình, sau khi thực thi yêu cầu xong, web service sẽ trả lại kết quả, cũng theo chuẩn SOAP.

Đọc XML

- Các đối tượng xử lý XML được cung cấp trong System.XML
 - Để đọc file XML, dùng lớp XmlTextReader
 - Hàm tạo với tham số là tên file XML cần đọc.
 - Phương thức Read() đọc 1 thành phần của file. Khi đọc thành công, đối tượng sẽ trở tới node hiện thời.
 - Tại một node, chúng ta có thể đọc được một số thành phần.
 - Kiểu của node: thuộc tính NodeType
 - Có những kiểu như Element, Text, CDATA, Comment,...
 - Giá trị của node: thuộc tính Value
 - Có thể sử dụng các phương thức cụ thể khác như ReadString(): đọc giá trị text của node, ReadAttributeValue(): đọc giá trị thuộc tính của node.

Đọc XML

```
public class Sample
{
    static void Main(string[] args)
    {
        XmlTextReader textReader = new XmlTextReader("C:\\books.xml");
        textReader.Read();
        while (textReader.Read() ) // If the node has value
        {
            // Move to first element
            textReader.MoveToElement();
            Console.WriteLine("XmlTextReader Properties Test");
            Console.WriteLine("=====");
            // Read this element's properties and display them on console
            Console.WriteLine("Name:" + textReader.Name);
            Console.WriteLine("Base URI:" + textReader.BaseURI);
            Console.WriteLine("Local Name:" + textReader.LocalName);
            Console.WriteLine("Attribute Count:" + textReader.AttributeCount.ToString());
            Console.WriteLine("Depth:" + textReader.Depth.ToString());
            Console.WriteLine("Line Number:" + textReader.LineNumber.ToString());
            Console.WriteLine("Node Type:" + textReader.NodeType.ToString());
            Console.WriteLine("Attribute Count:" + textReader.Value.ToString());
        }
    }
}
```


Ghi XML

■ Để ghi XML, dùng XmlTextWriter.

- Hàm tạo với tên file cần ghi.
- Các phương thức quan trọng.
 - WriteStartDocument(): ghi phần khai báo version XML.
 - WriteStartElement(string): ghi thẻ mở đầu của một nút.
 - WriteAttributeString(string, string): ghi thuộc tính và giá trị của nó.
 - WriteElementString(string, string): ghi một nút, trong đó có chứa một giá trị.
 - WriteEndElement(): ghi thẻ kết thúc của một nút.

Ghi XML

```
public class Sample
{
    public static void Main()
    {
        XmlTextWriter writer = new XmlTextWriter("titles.xml", null);

        //Write the root element
        writer.WriteStartElement("items");

        //Write sub-elements
        writer.WriteElementString("title", "Unreal Tournament 2003");
        writer.WriteElementString("title", "C&C: Renegade");
        writer.WriteElementString("title", "Dr. Seuss's ABC");

        // end the root element
        writer.WriteEndElement();

        //Write the XML to file and close the writer
        writer.Close();
    }
}
```

DOM trong .NET

- DOM (Document Object Model): mô hình đối tượng tài liệu cho phép xử lý XML một cách mềm dẻo
 - Khác với XmlTextReader, XmlTextWriter chỉ cho phép đọc và ghi XML theo kiểu tuần tự, DOM cho phép truy cập ngẫu nhiên vào tài liệu XML.
 - Các lớp quan trọng
 - XmlDocument: cho phép xử lý XML theo DOM
 - XmlNodeList: danh sách các node trong tài liệu XML.
 - XmlNode: một node đơn nhất trong tài liệu
 - XmlDocument cho phép thêm node mới, nối node vào đuôi tài liệu, xóa node khỏi tài liệu.

DOM trong .NET

```
void Sample()
{
    oXmlDoc = new XmlDocument();
    oXmlDoc.Load(Server.MapPath("xml_XmlDocument.xml"));
    XmlNode oNode = oXmlDoc.DocumentElement;
    Console.WriteLine("Node Name: " + oNode.Name);
    XmlNodeList oNodeList = oNode.SelectNodes("/books/category/title");
    Console.WriteLine("NodeList count=" + oNodeList.Count);
    for(int x = 0; x < oNodeList.Count; x++)
        Console.WriteLine("NodeList Item#" + x + " " +
            oNodeList.Item[x].InnerText);
}
```

Ứng dụng

- Tài liệu XML có thể dùng lưu trữ dữ liệu như là một CSDL.
- Một ứng dụng:
 - Xây dựng chương trình thi trắc nghiệm qua mạng, trên nền web:
 - Hiển thị câu hỏi, danh sách các đáp án
 - Số lượng đáp án trong một câu hỏi có thể khác nhau, và không hạn chế.
 - Một câu hỏi chỉ có một đáp án đúng.

Ứng dụng

■ Phân tích

- Các yêu cầu đầu bài.
- Thiết kế CSDL dưới dạng file XML
 - Đề ra một mô hình lưu trữ phù hợp
 - Tạo thử file dữ liệu
- Design form
- Viết mã dựa trên phân tích yêu cầu

Tài liệu tham khảo

- Professional C#, Second Edition
- <http://www.asp.net>
- <http://www.microsoft.com/net/default.mspix>
- <http://www.codeproject.com>
- Địa chỉ download tài liệu
<http://www.thanglong.edu.vn/giang-day/tab.aspx>
- Diễn đàn C# & .NET
<http://www.thanglong.edu.vn/forum/cmd/0/category/hoc-tap-nghien-cuu/dot-net/tab.aspx>