



XML VÀ .NET VỚI C#

PHAN TRỌNG TIẾN
BM Công nghệ phần mềm
Khoa Công nghệ thông tin, VNUA
Email: phantien84@gmail.com
Website: <http://timoday.edu.vn>



Mục tiêu

- ☐ Hiểu được XML là gì? Và ứng dụng XML.
- ☐ XML và hỗ trợ trong .NET
- ☐ Hiểu được các mô hình phân tích cấu trúc XML
- ☐ Giới thiệu thư viện XML.NET



Nội dung chính

- ☐ Tổng quan về XML
- ☐ XML và Microsoft .NET
- ☐ Các mô hình phân tích cấu trúc XML
- ☐ Thư viện XML.NET
- ☐ Sinh code tự động XML trong .NET
- ☐ Kiểm tra tính hợp lệ XML trong .NET
- ☐ Áp dụng kiểu style sheet cho các tài liệu XML

7/5/16

XML và .NET với C#

3



Tổng quan XML

- ☐ XML (EXtensible Markup Language) được thiết kế để miêu tả dữ liệu
- ☐ Là ngôn ngữ đánh dấu giống như HTML
- ☐ Các thẻ (tags) là tự định nghĩa bởi người sử dụng
- ☐ Được dùng như một giao thức tích hợp giữa các ứng dụng
- ☐ Tiêu chuẩn công nghiệp được phát triển bởi W3C

7/5/16

XML và .NET với C#

4



Ví dụ một file XML

```
<?xml version="1.0" encoding="UTF-8"?>

<PRODUCTDATA>
  <PRODUCT PRODID="P001">
    <PRODUCTNAME>Windows XP SP2</PRODUCTNAME>
    <DESCRIPTION>
      This is the latest update pack provided by Microsoft.
    </DESCRIPTION>
    <DETAILS>Has Firewall Protection</DETAILS>
    <PRICE>Free</PRICE>
    <SIZE>~200 MB for Windows XP Home Edition</SIZE>
  </PRODUCT>
</PRODUCTDATA>
```

7/5/16

XML và .NET với C#

5



Cấu trúc XML

```
<?xml version="1.0" encoding="UTF-8"?>
<PRODUCTDATA>
  <PRODUCT PRODID="P001">
    <PRODUCTNAME>Windows XP SP2</PRODUCTNAME>
    <DESCRIPTION>
      This is the latest update pack provided by Microsoft.
    </DESCRIPTION>
    <DETAILS>Has Firewall Protection</DETAILS>
    <PRICE>Free</PRICE>
    <SIZE>~200 MB for Windows XP Home Edition</SIZE>
  </PRODUCT>
</PRODUCTDATA>
```

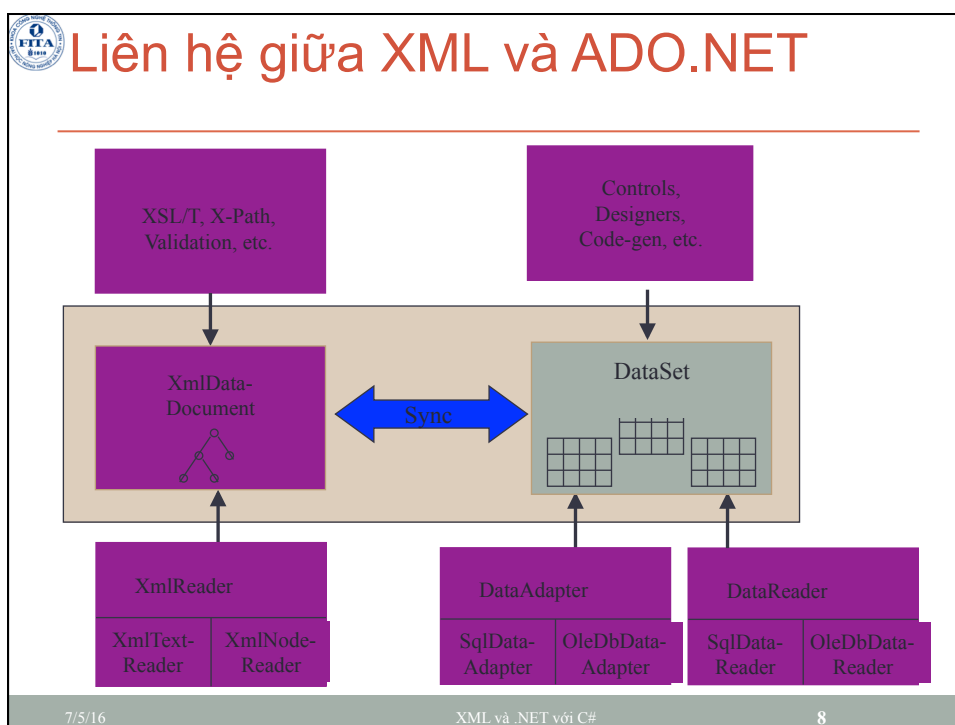
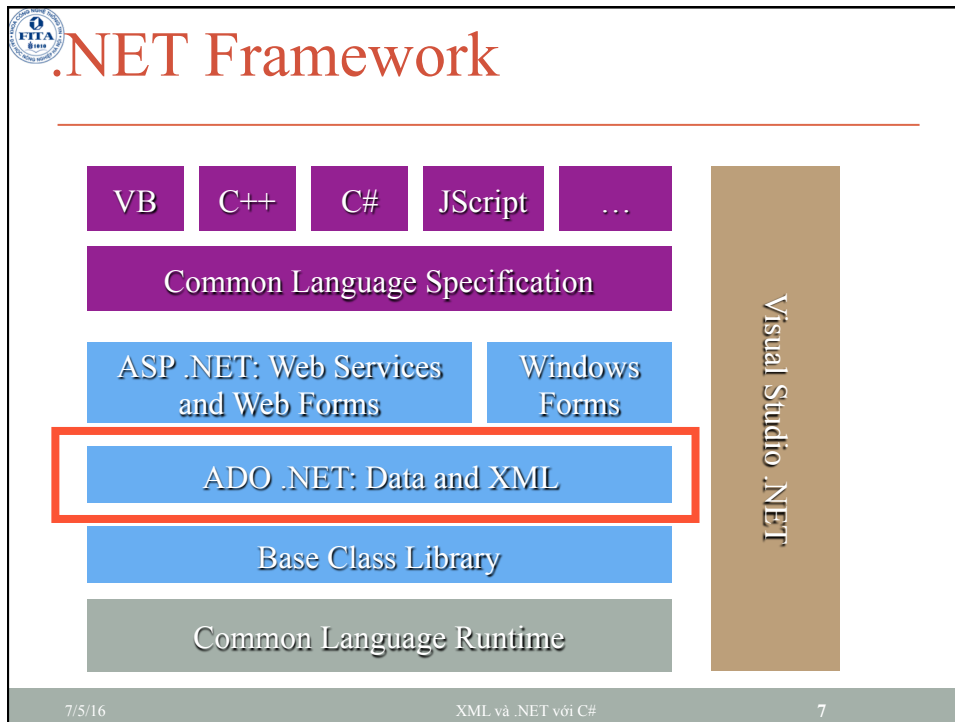
Diagram illustrating XML structure components:

- Declaration:** Points to `<?xml version="1.0" encoding="UTF-8"?>`
- Root Element:** Points to `<PRODUCTDATA>`
- Parent Element:** Points to `<PRODUCT PRODID="P001">`
- Attribute:** Points to `PRODID="P001"`
- Child Element:** Points to `<PRODUCTNAME>Windows XP SP2</PRODUCTNAME>`
- Content:** Points to the text `This is the latest update pack provided by Microsoft.`

7/5/16

XML và .NET với C#

6





XML và .NET

- ❑ XML: là một chuẩn công nghiệp, là kỹ thuật kết nối các ứng dụng
- ❑ .NET: là nền tảng công nghệ của MS nhằm phát triển các ứng dụng phân tán
- ❑ XML là chất gắn kết giữa các thành phần (components) trong .NET lại với nhau

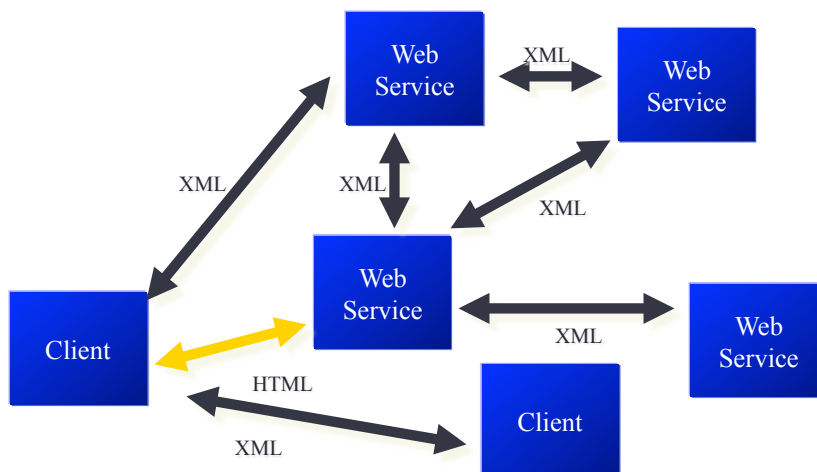
7/5/16

XML và .NET với C#

9



XML.NET và Web Services



7/5/16

XML và .NET với C#

10



Các mô hình phân tích XML

- ❑ Các mô hình phân tích XML thông dụng
 - ❑ Sử dụng mô hình DOM (Document Object Model)
 - ❑ Mô hình SAX (Simple API for XML): sử dụng mô hình đẩy dữ liệu (push model parsing)
 - ❑ Mô hình phân tích kéo dữ liệu (pull model parsing)
- ❑ .NET Framework hỗ trợ hai mô hình
 - ❑ Mô hình pull
 - ❑ Mô hình DOM
- ❑ Những thuận lợi và giới hạn của mỗi mô hình?

7/5/16

XML và .NET với C#

11



Mô hình DOM

- ❑ Phân tích XML trong bộ nhớ
 - ❑ Một cây cấu trúc được tạo trong bộ nhớ, mô tả nội dung của tài liệu XML
- ❑ Mô hình này dùng khi có bạn cần duyệt động (dynamically navigate) và thực thi insert, update và delete nội dung trong tài liệu XML
- ❑ Không thích hợp khi bạn chỉ cần phân tích cấu trúc XML từ đầu tới cuối theo kiểu read-only
- ❑ Ảnh hưởng bộ nhớ: Nạp một tệp XML lớn vào trong DOM có thể làm cạn kiệt tài nguyên

7/5/16

XML và .NET với C#

12



Phân tích sử dụng DOM

```
<?xml version="1.0"?>
<Books>
  <Book ISBN="0355605172">
    <Title>Beginning XML</Title>
    <Price>40.00</Price>
  </Book>
  <Book ISBN="0415205173">
    <Title>XML Step by Step</Title>
    <Price>50.00</Price>
  </Book>
</Books>
```

7/5/16

XML và .NET với C#

13



DOM parsing — truy cập và chỉnh sửa một phần dữ liệu

```
static void Main(string[] args)
{
    XmlDocument xmlDoc = new XmlDocument();
    xmlDoc.Load("books.xml");
    XmlNodeList priceNodes = xmlDoc.GetElementsByTagName("Price");
    double price;
    foreach(XmlNode priceNode in priceNodes)
    {
        price = Convert.ToDouble(priceNode.InnerText);
        if (price >= 50)
        {
            price = price - (5 * price / 100);
            //Cập nhật nội dung vào the
            priceNode.InnerText = price.ToString();
        }
    }
    //Lưu nhưng thay đổi vào file
    xmlDoc.Save("books.xml");
}
```

7/5/16

XML và .NET với C#

14



Lỗi namespace của XML.NET

- ❑ System.Xml
 - ❑ Là namespace cung cấp toàn bộ các lớp chuẩn (standards-based) hỗ trợ phân tích (parsing) XML
 - ❑ Hỗ trợ các chuẩn
 - ❑ XML 1.0 và các namespace XML
 - ❑ XML scheme
 - ❑ Xpath
 - ❑ XSLT
 - ❑ DOM mức 2
 - ❑ SOAP 1.1 (được sử dụng trong các đối tượng tuần tự)

7/5/16

XML và .NET với C#

15



Lỗi namespace của XML.NET

- ❑ System.Xml.Xsl
 - ❑ Chứa các lớp hỗ trợ biến đổi XSLT
- ❑ System.Xml.XPath
 - ❑ Chứa các lớp hỗ trợ thực thi các truy vấn XPath
- ❑ System.Xml.Schema
 - ❑ Chứa các lớp chuẩn hỗ trợ cho W3C XML schemas
- ❑ System.Xml.Serialization
 - ❑ Chứa các lớp được dùng cho tuần tự hoặc không tuần tự các đối tượng XML

7/5/16

XML và .NET với C#

16



Tự động phát sinh XML trong .NET

- ☐ Các tùy chọn có sẵn để lập trình với XML
 - ☐ Non-cached, forward-only streaming
 - ☐ Lập trình với DOM
- ☐ Các thuận lợi và giới hạn của mỗi phương thức

7/5/16

XML và .NET với C#

17



Sử dụng lớp XmlTextWriter

- ☐ Thực thi trong namespace System.Xml
- ☐ Thừa kế từ lớp trừu tượng System.Xml.XmlWriter
- ☐ Được sử dụng lập trình phát sinh XML non-cached, kiểu forward-only
- ☐ Có thể được dùng để phát sinh XML tới một file trên disk và các đối tượng .NET Framework Stream/TextWriter

7/5/16

XML và .NET với C#

18



Sử dụng lớp XmlTextWriter

```
<?xml version="1.0"?>
<!--Catalog fragment-->
<!DOCTYPE Books SYSTEM "books.dtd">
<Books>
  <Book ISBN="0355605172">
    <Title>XML Step by Step</Title>
  </Book>
</Books>
```

7/5/16

XML và .NET với C#

19



Sử dụng lớp XmlTextWriter

```
XmlTextWriter wrt = new XmlTextWriter("books.xml",null);
wrt.Formatting = System.Xml.Formatting.Indented;

wrt.WriteStartDocument(false);
wrt.WriteComment("Catalog fragment");
wrt.WriteDocType("Books", null, "books.dtd", null);

wrt.WriteStartElement("Books");

wrt.WriteStartElement("Book");
wrt.WriteAttributeString("", "ISBN", "", "0355605172");

wrt.WriteStartElement("Title");
wrt.WriteString("XML Step by Step");
wrt.WriteEndElement();

wrt.WriteEndElement();

wrt.WriteEndElement();

wrt.Close();
```

7/5/16

XML và .NET với C#

20



Sử dụng DOM

```

XmlDocument xmldoc = new XmlDocument();
XmlDeclaration xmlDecl = xmldoc.CreateXmlDeclaration("1.0", null, null);
xmldoc.AppendChild(xmlDecl);

XmlDocumentType docType = xmldoc.CreateDocumentType("Books", null,
"c:\books.dtd", null);
xmldoc.AppendChild(docType);

XmlComment xmlComment = xmldoc.CreateComment("Catalog fragment");
xmldoc.AppendChild(xmlComment);

xmldoc.AppendChild(xmldoc.CreateElement("Books"));

xmldoc.DocumentElement.AppendChild(GenerateBookNode(xmldoc, "XML
Step by Step", "0355605172"));
xmldoc.Save("c:\books2.xml");

```

7/5/16

XML và .NET với C#

21



Sử dụng DOM

```

private static XmlNode GenerateBookNode(XmlDocument xmldoc,
String Title, String ISBN)
{
    XmlNode bookNode;

    bookNode = xmldoc.CreateElement("Book");
    bookNode.AppendChild(xmldoc.CreateElement("Title"));
    bookNode.ChildNodes[0].InnerText = Title;

    bookNode.Attributes.Append(xmldoc.CreateAttribute("ISBN"));
    bookNode.Attributes.GetNamedItem("ISBN").InnerText =
ISBN;

    return bookNode;
}

```

7/5/16

XML và .NET với C#

22



7/5/16 XML và .NET với C# 23

Kiểm tra tính hợp lệ của XML

- Các lược đồ (schema) miêu tả cấu trúc của file XML. Kiểm tra tính hợp lệ đảm bảo rằng mở rộng dữ liệu phù hợp với các quy tắc (ngữ pháp) được yêu cầu bởi lược đồ

```
<?xml version="1.0"?>
<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema>
<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

7/5/16 XML và .NET với C# 24



Kiểm tra tính hợp lệ của XML

- ❑ Có 3 ngôn ngữ được đề nghị:
 - ❑ Document Type Definitions (DTD)
 - ❑ XML Data Reduced schema (XDR)
 - ❑ XML Schema Definition language (XSD)
- ❑ XSD là tương lai. Các lược đồ có nhiều lợi thế hơn DTD:
 - ❑ Lược đồ sử dụng câu lệnh XML và có thể được dùng bởi một phân tích XML
 - ❑ Lược đồ hỗ trợ các kiểu dữ liệu (integer, string, Boolean) và tương thích với các kiểu dữ liệu khác

7/5/16

XML và .NET với C#

25



Lược đồ trong .NET

- ❑ Dữ liệu XML có thể được xác nhận (validate) cả 3 ngôn ngữ lược đồ sử dụng trong các lớp .NET
- ❑ System.Xml.XmlValidatingReader được dùng cho sự xác nhận
- ❑ System.Xml.Schema cung cấp chuẩn cơ sở cho lược đồ XML (cấu trúc và kiểu dữ liệu)
- ❑ System.Xml.Schema.XmlSchemaCollection chứa một cache của lược đồ XSD và XDR

7/5/16

XML và .NET với C#

26



Các biến đổi XSLT trong .NET

- ❑ XSL (Extensible Stylesheet Language) bao gồm 3 phần:
 - ❑ XSLT – XSL transformation
 - ❑ Xpath – XML path language
 - ❑ XSL-FO – XSL formatting objects
- ❑ XSLT là một ngôn ngữ để chuyển đổi các tài liệu XML thành các tài liệu thuần văn bản (text-based)
- ❑ Quá trình chuyển đổi bao gồm 3 loại:
 - ❑ File XML nguồn
 - ❑ File Stylesheet
 - ❑ File output: xml, html, v.v

7/5/16

XML và .NET với C#

27



XSLT trong .NET

- ❑ Thực thi dưới namespace System.Xml.Xsl
- ❑ Hỗ trợ theo khuyến nghị W3C XSLT 1.0
- ❑ Các lớp
 - ❑ **XslTransform** – biến đổi dữ liệu XML sử dụng một XSLT stylesheet
 - ❑ **XslArgumentList** – Cho phép các tham biến và các đối tượng được gọi trong stylesheet
 - ❑ **XsltException** – Trả về thông tin về exception trong khi xử lý một XSLT

7/5/16

XML và .NET với C#

28



Tổng quan XPath

- ☐ Là một truy vấn (query) cho XML – SQL của XML
- ☐ Được sử dụng để xác định các biểu thức truy vấn tới vị trí các nút trong tài liệu XML
- ☐ Được sử dụng trong XSLT stylesheets để xác định vị trí và áp dụng biến đổi tới các nút cụ thể trong một tài liệu XML
- ☐ Được sử dụng DOM code để xác định vị trí và xử lý các nút cụ thể

7/5/16

XML và .NET với C#

29



XPath trong .NET

- ☐ Các namespace liên quan
 - ☐ System.Xml
 - ☐ System.Xml.XPath
- ☐ Các lớp trong .NET Framework
 - ☐ XmlDocument, XmlNodeList và XmlNode
 - ☐ XPathDocument
 - ☐ XPathNavigator
 - ☐ XPathNodeIterator
 - ☐ XPathExpression

7/5/16

XML và .NET với C#

30



Thực thi các truy vấn XPath

- ❑ **Sử dụng đối tượng DOM của System.Xml**
 - ❑ Các class thường được sử dụng: **XMLDocument**, **XMLNodeList**, **XmlNode**
 - ❑ Nạp tài liệu XML vào trong lớp **XMLDocument**
 - ❑ Sử dụng phương thức **SelectNodes/SelectSingleNode** của lớp **XmlDocument** để thực thi truy vấn XPath
 - ❑ Khởi gán trả về danh sách các node/node tới đối tượng **XmlNodeList/XmlNode**
 - ❑ Sử dụng đối tượng **XmlNode** để duyệt qua **XmlNodeList** và xử lý kết quả

7/5/16

XML và .NET với C#

31



Thực hiện truy vấn XPath

- ❑ Ví dụ file XML “books.xml”


```
<?xml version="1.0"?>
<Books>
  <Book ISBN="0355605172">
    <Title>The C Programming language</Title>
  </Book>
  <Book ISBN="0735605173">
    <Title>XML Step by Step</Title>
  </Book>
</Books>
```
- ❑ Ví dụ cần thực hiện truy vấn lấy tất cả tên những quyển sách bắt đầu bằng từ “XML”
 - ❑ **Truy vấn XPath:** `//Title[starts-with(., 'XML')]`

7/5/16

XML và .NET với C#

32



Thực thi truy vấn XPath

```
static void Main(string[] args)
{
    XmlDocument xmlDoc = new XmlDocument();
    xmlDoc.Load("books.xml");
    XmlNodeList matchingNodes = xmlDoc.SelectNodes("//Title[starts-
with(.,'XML')]");
    if (matchingNodes.Count == 0)
    {
        Console.WriteLine("No matching nodes were identified for the specified XPath
query");
    }
    else
    {
        foreach (XmlNode node in matchingNodes)
        {
            Console.WriteLine(node.Name + ":" + node.InnerText);
        }
    }
    Console.ReadLine();
}
```

7/5/16

XML và .NET với C#

33



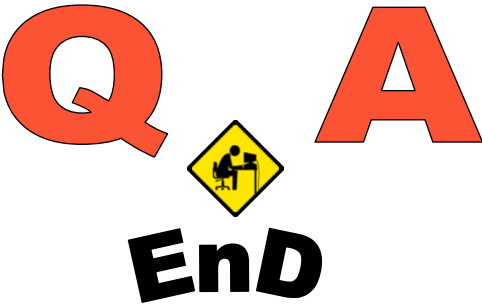
Tổng kết

- ☐ Đã giới thiệu XML trong .NET
- ☐ Đã trình bày *các mô hình phân tích XML* trong .NET Framework
- ☐ Nghiên cứu về **phân tích DOM** trong .NET Framework
- ☐ **Các namespace .NET** liên quan trong XML
- ☐ Thực hiện lập trình **tạo ra XML** trong .NET
- ☐ Thực hiện lập trình **kiểm tra hợp lệ các tài liệu XML**
- ☐ Thực thi **biến đổi XSLT**
- ☐ Thực thi **truy vấn XPath**

7/5/16

XML và .NET với C#

34



Q A

EnD



7/5/16 XML và .NET với C# 35



Tài liệu tham khảo

- ❑ <http://www.w3schools.com/xml/>
- ❑ <http://web2.uwindsor.ca/courses/cs/aggarwal/HPGCCGroup/2.ppt>
- ❑ <http://www.firstobject.com/xml-reader-sax-vs-xml-pull-parser.htm>
- ❑ <https://support.microsoft.com/en-us/kb/317463>

7/5/16 XML và .NET với C# 36