田 原 唐铸文 著

XML与Visual Basic.NET编程技术

科学出版社
www.sciencep.com

(TP-3277.0101)

XML与Visual Basic.NET编程技术

ISBN 7-03-017379-1
定 价：42.00 元

XML 与 Visual Basic .NET 编程技术

田 原 唐铸文 著

科学出版社

北京

内 容 简 介

XML 是近年来热门的 IT 技术之一，并且已经广泛应用于编程领域中。本书包括两部分：XML 技术基础和 XML 与 Visual Basic .NET 编程技术。第 1 部分主要内容包括 XML 概述、XML 文档示例、使用 XML 创建标记、文档类型定义、Schema、样式表、XML 协议，以及 DOM 与 SAX 接口。第 2 部分主要内容包括 Visual Basic .NET 基础、在 .NET 中读取 XML、在 .NET 中编写 XML、在 .NET 中实现 DOM、在 .NET 中使用 XSLT、XML 架构和 .NET、XML 和 ADO.NET，以及 XML Web 服务。

本书结构清晰、内容丰富、重点突出、语言流畅，既有理论性的阐述，也有具体的开发实例。本书适合具有一定 Visual Basic .NET 编程经验，并利用 XML 与 Visual Basic .NET 进行编程开发的技术人员。

图书在版编目（CIP）数据

XML 与 Visual Basic .NET 编程技术/田原，唐铸文著. —北京：科学出版社，2006

ISBN 7-03-017379-1

Ⅰ.X… Ⅱ.①田… ②唐… Ⅲ. ①可扩充语言，XML—程序设计 ②BASIC 语言—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2006）第 059828 号

责任编辑：李 伟/责任校对：刘彦妮

责任印制：吕春珉/封面设计：耕者设计工作室

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2006 年 6 月第 一 版 开本：787×1092 1/16

2006 年 6 月第一次印刷 印张：21 1/4

印数：1—700 字数：487 000

定价：42.00 元

（如有印装质量问题，我社负责调换〈环伟〉）

销售部电话 010-62136131 编辑部电话 010-62135763-8220

前　言

XML 是近年来热门的 IT 技术之一，并且已经广泛应用于编程领域中。本书包括两部分：XML 技术基础和 XML 与 Visual Basic .NET 编程技术。主要内容介绍如下。

第 1 部分　XML 技术基础

第 1 章：XML 概述。本章介绍 XML 的基本知识，概略地解释什么是 XML 及如何使用 XML，并说明如何将各种不同的 XML 表达式组合在一起，XML 文档是如何创建的，以及如何向人们发送这种文档。

第 2 章：第一个 XML 文档。本章将用用户自己定义的可被文档所理解的标记来创建一个简单的 XML 文档。简要介绍如何编写样式单，以便用于在文档中描述标记内容如何显示。最后，还介绍了如何将文档装到 Web 浏览器中以便查看。

第 3 章：使用 XML 创建标记。本章介绍 XML 标记的格式，创建规范，XML 分析程序，以及如何使用 msxml 分析 XML 文档。

第 4 章：文档类型定义（Document Type Definition，DTD）。本章讨论 DTD，它可以定义 XML 文档的结构（例如文档中有什么元素、属性等）。XML 文档要求有一个对应的 DTD，用于确保文档的一致性。

第 5 章：架构（Schema）。本章讨论 DTD 的一个替代产品——Schema，并用它来检查 XML 文档的有效性。与 DTD 一样，Schema 必须与验证有效性的分析程序一起使用。Schema 能够替代 DTD 的主要原因是它能够作为描述文档结构的主要方式。

有两种重要的 Schema 模型：Microsoft XML Schema 和 W3C XML Schema。因为 W3C XML Schema 技术仍处于早期开发阶段，所以本章主要讨论已开发成熟的 Microsoft XML Schema。

第 6 章：样式表。对于一批 XML 数据，应用处理程序要综合 XML 文档、文档类型说明及样式表三方面要素来处理和显示它。层叠样式单 CSS 和扩展样式表语言 XSL 是两种重要的样式表，本章介绍了它们的功能，并以示例的形式介绍了它们的使用方法。

第 7 章：XML 协议。本章介绍 XML 协议，包括 XML 的基础标准、SGML 标准体系框架、XML 标准体系框架及相关应用协议。

第 8 章：DOM（Document Object Model，文档对象模型）与 SAX（Simple API for XML，XML 简单应用程序接口）接口。本章讨论由 W3C 和 XML_DEV 邮件列表成员分别提出的两个标准应用程序接口：DOM 和 SAX。

第 2 部分　XML 与 Visual Basic .NET 编程技术

第 9 章：Visual Basic .NET 基础。本章简单介绍了 .NET Framework 和 Visual Basic .NET 的一些新的特性，通过本章学习，读者应该对 .NET Framework 和 Visual Basic .NET 的集成开发环境有一定的了解，为后续开发打下基础。

第 10 章：在 .NET 中读取 XML。本章主要讨论如何利用 Visual Basic .NET 来读取 XML。将研究如何通过遍历 XML 文档读取不同的节点类型，并考虑如何读取二进制数据和较大的 XML 文档，而且还要讨论如何实现在 XML 文档中的一些验证。

第 11 章：在 .NET 中编写 XML。本章讨论如何通过编程的方式，利用 .NET 类中定义的功能把 XML 数据写入到 Visual Basic .NET 程序代码中。

第 12 章：在 .NET 中实现 DOM。本章将讨论如何在 .NET Framework 中实现文档对象模型。具体讨论的内容有：DOM 的定义以及它与流模型的区别；使用 XmlNode 类；使用 XmlDocument 类。

第 13 章：在 .NET 中使用 XSLT。本章讨论了如何使用 .NET 类来实现 XSL。主要内容有：XslTransform 类；如何进行一些简单的 XSLT 转换，将其转换到文件或内存中；.NET 如何转换样式表；如何支持 Visual Basic .NET 脚本；通过参数和外部对象使 XSLT 与环境进行交互。

第 14 章：XML 架构和 .NET。在本章中，把 XML 架构的有关知识扩展到 .NET 中。本章讨论的主要内容：如何在 Visual Studio .NET 中编辑架构；通过编程方式验证 XML；XSD 和利用 xsd.exe 实用程序进行串行化。

第 15 章：XML 和 ADO .NET。本章首先讨论一些基本的技术，以说明如何利用各种 ADO .NET 类执行普通的数据访问任务。最后讨论了 XmlDataDocument 类，说明如何利用它同时查看并处理一个分层的关系式数据表示，并为用户提供了利用 DOM 定位关系数据的方法，以及利用 XPath 表达式查询和应用 XSLT 转换的能力。

第 16 章：XML Web 服务。在 Visual Basic .NET 中可以快速地建立起 Web 服务，可以很容易地开发出客户端的应用程序。在本章中，将讨论如何在客户端使用这些 Web 服务，如何利用 Web 服务来满足用户对信息的需求。

本书由田原、唐铸文共同撰写。其中第 1～8 章由唐铸文撰写，第 9～16 章由田原撰写。田原负责审定全书内容。

由于作者水平有限，书中不妥或错误之处在所难免，敬请读者指正。

目　录

第 1 部分　XML 技术基础

第 2 部分　XML 与 Visual Basic .NET 编程技术

第 1 部分　XML 技术基础

- XML 概述
- 第一个 XML 文档
- 使用 XML 创建标记
- 文档类型定义
- Schema
- 样式表
- XML 协议
- DOM 与 SAX 接口

第 1 章　XML 概述

本章将介绍 XML 的基本知识并概略地解释什么是 XML 及如何使用 XML。还要说明如何将各种不同的 XML 表达式组合在一起，如何创建 XML 文档，以及如何向用户发送这种文档。

1.1　什么是 XML

XML（eXtensible Markup Language，可扩展的标记语言）是一套定义语义标记的规则，这些标记将文档分成许多部件并对这些部件加以标识。它也是元标记语言，即定义了用于定义其他与特定领域有关的、语义的、结构化的标记语言的句法语言。

1.1.1　XML 是元标记语言

XML 与超文本标记语言（Hypertext Markup Language，HTML）或格式化程序相比有一定优势。HTML 等标记语言定义了一套固定的标记，用来描述一定数目的元素。如果标记语言中没有所需的标记，用户也就没有办法了。这时只好等待标记语言的下一个版本，希望在新版本中能够包括所需的标记，但是这样一来就得依赖于软件开发商了。

XML 是一种元标记语言，用户可以定义自己需要的标记。这些标记必须根据某些通用的原理来创建，但是在标记的意义上具有相当的灵活性。例如，假如用户正在处理与通讯录有关的事情，需要描述人的姓名、工作单位、通讯地址、各种电话号码、E-mail 等，就必须创建用于每项的标记。新创建的标记可在 DTD 中加以描述。在本书的后面将介绍有关 DTD 的更多知识。现在，只需把 DTD 看作是一本词汇表和某类文档的句法。

XML定义了一套元句法，与特定领域有关的标记语言（如MusicML、MathML和CML）都必须遵守。如果一个应用程序可以理解这些元句法，那么它也就能够自动地理解所有的由此元语言建立起来的语言。浏览器不必事先了解多种不同的标记语言使用的每个标记。事实是，浏览器在读入文档或是它的DTD时才了解了给定文档使用的标记。显示这些标记内容的详细指令是由附加在文档上的另外的样式表提供的。

有了XML就意味着不必等待浏览器开发商来满足用户的需要，用户可以创建自己需要的标记，当需要时，告诉浏览器如何显示这些标记即可。

1.1.2 XML描述的是结构和语义，而不是格式化

XML标记描述的是文档的结构和意义，它不描述页面元素的格式化。可用样式表为文档增加格式化信息。文档本身只说明文档包括什么标记，而不是说明文档看起来是什么样的。

比较而言，HTML文档中的标记包括了格式化、结构和语义信息。<B>就是一种格式化标记，它使其中的内容变为粗体。<STRONG>是一种语义标记，意味着其中的内容特别重要。<TABLE>是一种结构标记，指明建立一个表格。事实上，某些标记可能具有所有这三种意义。<H1>标记可同时表示20磅的Helvetica字体的粗体、第一级标题和页面标题。

例如，在HTML中，一本书可能是用定义标题、数据、无序的列表和列表项来描述的。但是事实上这些项目没有一件是与书有关的。用HTML定义的书可能如下：

```
<dt>人气
<dd>蒋子龙
<ul>
    <li>出版社：作家出版社
    <li>字数：411千字
    <li>出版时间：1999年12月
    <li>书号：7-50063-1764-8/I.1752
</ul>
```

而在XML中，同样的数据可能标记为：

```
<BOOK>
   <BOOKNAME>人气</BOOKNAME>
   <AUTHOR>蒋子龙</AUTHOR>
   <PUBLISHER>作家出版社</PUBLISHER>
   <LENGTH>411千字</LENGTH>
   <DATE>1999年12月<DATE>
   <BOOKNO>7-50063-1764-8/I.1752<BOOKNO>
</BOOK>
```

在这个清单中没有使用通用的标记如<dt>和<li>，而是使用了具有一定意义的标记，如<BOOKNAME>、<AUTHOR>、<PUBLISHER>等。这种用法具有许多优点，如源码易于阅读，使人能够看出作者的含义。

XML 标记还使非人类的自动机器人易于找出文档中的所有书。在 HTML 中，机器人只能告诉我们这个元素是 dt，而不能确定 dt 到底代表一本书的题目还是定义，抑或只是一些设计者喜爱的缩进文本格式。事实上，单一文档中可以很好地包括带有三种意义的各种 dt 元素。

可以选择 XML 的元素名称，以便使其在附加的上下文中具有额外的意义。例如，元素名称可以是数据库的域名。XML 比 HTML 更为灵活而且适用于各种应用，因为有限数目的标记不必用于许多不同的目的。

1.2　XML 的应用

XML 使许多只利用 HTML 难以解决的任务变得简单，使只利用 HTML 不可能完成的任务得以完成。因为 XML 是可扩展的。开发人员喜爱 XML 有许多原因。但到底是哪个更令人感兴趣，取决于每个人的需要。但有一点是肯定的，一旦用上 XML，就可发现，它正是解决许多令人感到棘手的问题的有力工具。本节研究一些令开发人员激动的一般应用。

1.2.1　设计与特定领域有关的标记语言

XML 允许各种不同的专业（如音乐、化学和数学等）开发与自己的特定领域有关的标记语言。这就使该领域中的人可以交换笔记、数据和信息，而不用担心接收端的人是否有特定的软件来创建数据。特定领域的开发人员甚至可以向本领域外的人发送文档，至少接收文档的人能够查看文档的内容。

更进一步说，为特别的领域创建标记语言不会产生“病件”（bloatware）或是对于本专业外的人来说产生不必要的复杂性。一般人也许不会对电力工程图感兴趣，但是电力工程师却对此感兴趣。一般人也许不需要在他的 Web 页面中包含乐谱，但是作曲家却要这样做。XML 让电力工程师描述他们的电路图，让作曲家写乐谱，而不会互相干扰。对于浏览器开发商来说，不需要对特定的领域提供特殊的支持，也不需要提供复杂的插件。这一点现在已经实现了。

1.2.2　自描述数据

过去 40 多年来的大多数计算机数据都丢失了，不是因为自然损害或是备份介质的磨损（虽然这也是一个问题，并且在 XML 中也没有解决），而只是因为没有人写出如何读取这些数据介质和格式的文档。十多年前的 5.25 英寸（1 英寸=2.54 厘米）软盘上的 Lotus 1-2-3 文档，在今天的大多数公司内都已经读不出来了。以不常用的格式保存的二进制数据，如 Lotus Jazz 也许会永远地消失了。XML 基本上使用的是非常简单的数据格式，可以用 100%的纯 ASCII 文本来书写，也可以用几种其他定义好的格式来书写。ASCII 文本是几乎不会“磨损”的。丢失一些字节甚至是相当多的字节，剩下的数据还是可以读取的。这就与许多格式形成了鲜明的对比，如压缩数据或是串行的 Java 对象，这些数据即使丢失一个字节，剩余的数据也不可读取了。

从高水平上来说，XML 是自描述的。假设在 23 世纪有一个信息考古学者，他在软

盘上发现了如下一大段经过时间的“冲刷”而保存下来的XML代码：

```
<PERSON ID="2000" SEX="男">
  <NAME>李大军</NAME>
  <BIRTH>
     <DATE>1935年2月6日</DATE>
  </BIRTH>
  <DEATH>
     <DATE>2004年6月21日</DATE>
  </DEATH>
</PERSON>
```

即使这个考古学家不熟悉XML，但假设他可以讲20世纪时的英语和汉语，那么就可以很好地了解名为李大军的人，此人出生在1935年2月6日，而死于2004年6月21日。事实上，即使数据中有一些空白或是损坏，还是可以得到这些信息。但专有格式的电子表格或是字处理程序的格式就不是这样。

更进一步说，XML有很好的规范文档向人们准确地说明如何来阅读XML数据，并且这些规范文档都是公开的。

1.2.3 应用程序间的数据交换

由于XML是非专有的，并易于阅读和编写，就使它成为在不同的应用程序间进行数据交换的理想格式。当前正在开发的一种格式是OFX（Open Financial Exchange，开放财务交换）格式。OFX是为个人财务程序如Microsoft Money和Quicken交换数据而设计的。数据可以在程序间来回交换，还可以与银行、经纪事务所和其他机构之间进行交换。

正如上面所讨论的一样，XML使用的是非专有的格式，不受版权、专利、商业秘密或是其他种类的知识产权的限制。XML的功能是非常强大的，对于人类或计算机程序来说，都容易阅读和编写，因而成为交换语言的首选。

使用XML而不是专有格式，人们就可以利用任何理解XML的工具来处理数据。还可以为不同的目的使用不同的工具，一个程序用来查看而另一程序用来编辑。XML使用户不必因为数据已经用专有格式编写好了或是接收数据的人只接收专有格式而限制在一个特定的程序上。

例如，许多出版商需要用Microsoft Word发稿，这就意味着大多数作者必须使用Word，即使他们更愿意使用WPS Office。因而这就使其他出版字处理软件的公司陷入困境，除非它们的软件能够读写Word文件。由于要想达到这个目的，就得让开发人员反向了解未载入文档的Word文件格式，这使得在时间和资源上的投资大增。大多数其他字处理软件具有有限的读写Word文件的能力，但是通常都会丢失图形、宏、样式、修订标记和其他重要的特性。问题就在于Word文档的格式是不公开的专有格式，而且还在不断地变化。这样Word就成为最后的胜利者，即使作者更喜爱其他更简单的程序。如果在XML中开发了一种通用的字处理格式，作者们就会使这个程序成为他们的首选程序。

1.2.4 结构化和集成的数据

XML对于大型和复杂的文档而言是理想的，因为数据是结构化的。这不仅使用

户可以指定一个定义了文档中的元素的词汇表，而且还可以指定元素之间的关系。例如，如果要将销售客户的地址一起放在 Web 页面上，这就需要有每个客户的电话号码和电子邮件地址。如果向数据库中输入数据，可确保没有漏下的字段。当没有数据输入时还可提供一个默认值。XML 也提供客户端的包括机制，可以根据多种来源集成数据并将其作为一个文档来显示。数据还可以马上进行重新排列。数据的各个部分可以根据用户的操作显示或隐藏。这在处理大型的信息仓库（如关系型数据库）时是极为有用的。

1.3　XML 文档的“生命”

从基本上来说，XML 是一种文档格式。它是一系列的关于 XML 文档看起来是什么样式的规则。与 XML 标准的符合程度有两种级别：第一级是结构完整性，第二级是正确性。

HTML 是设计用于 Internet 上和 Web 页面内部的文档格式。正如本书所叙述的，XML 当然也可以用在这些方面，但是 XML 具有更为广泛的适用性，例如可用于设置字处理器保存文件的格式，可用于不同程序间的数据交换格式，可用作与 Intranet 模板一致化的工具，还可用作以人类可读的形式保存数据的手段。

虽然如此，如果所有的数据格式一样，XML 在使用之前也需要程序和内容。因而对于数据如何显示，光了解 XML 本身还是不够的，这不光是一个规范所能解决的问题。用户还需要了解 XML 文档是如何编辑的，处理程序是如何读取 XML 文档并将其读取的信息传送给应用程序的，以及这些应用程序是如何处理数据的。

1.3.1　编辑器

XML 文档大多数情况下都是用编辑器创建的。编辑器可以是基本的文本编辑器，如 Notepad（记事本）或是 DOS 下的 Edit，这些编辑器并不真正理解 XML。另一方面，也可以用所见即所得的编辑器，如 Adobe FrameMaker，这种编辑器可将用户完全隔离于 XML 底层格式之外。另外也可以是一个结构化的编辑器，如 JUMBO，它可将 XML 文档显示为树状结构。对于最重要的部分，有趣的编辑器并不是太有用，因而本书将注意力集中于用普通的文本编辑器来编写 XML 文档。

其他程序也可以创建 XML 文档。例如，本书在讲述设计新的 DTD 章节中将介绍某些 XML 数据可直接从 FileMaker 的数据库中得出。在这种情况下，数据是先输入到 FileMaker 数据库中的，然后 FileMaker 的计算字段将数据转换为 XML。一般来说，XML 与数据库可很好地协同工作。

无论在何种情况下，都是编辑器或其他程序创建了 XML 文档。通常，XML 文档是某种计算机硬盘上的实际文件。但也不是必须如此，例如，文档可能是数据库中的记录或是字段，或者可能是从网络上接收的字节流。

1.3.2　语法分析程序

XML 的语法分析程序（即 XML 处理程序）读取文档并检查其中包括的 XML 是否是结构完整的。它还要确定文档是否合法，虽然这种测试不是必需的。这种测试的详细