XML for ASP.NET Dev

基于 XML 的 ASP.NET 开发

SAMS

基于 XML 的 ASP.NET 开发

XML for ASP.NET Developers

[美] Dan Wahlin 著

王宝良 译

清 华 大 学 出 版 社

（京）新登字 158 号

著作权合同登记号：01-2002-3071

内 容 简 介

随着微软引入了.NET 平台，XML 在编程领域中的应用达到了新的高潮。本书详细介绍了这种前途无量的标记语言，全面展示了 ASP.NET 开发人员如何使用 XML 来提高应用程序的效率。

本书首先介绍了一些不可不知的概念，包括 XML 元数据语言、文档类型定义(DTD)、XML Schema、XPath 语言，以及需要留意的最新技术——XPointer 和 XLink。然后深入探讨了如何以多种不同的方法使用 ASP.NET 来创建、解析和转换 XML 文档，并将其传送给其他系统。第 2 部分是本书的精华所在，为了鲜活地体现这一点，书中提供了大量实例代码，帮助读者轻松掌握相关的概念并马上投入实践。

本书适用于广大网站开发人员、应用程序开发员、产品技术员，以及.NET 爱好者。对 ASP.NET 开发人员来说，本书更是必不可少的。

XML for ASP.NET Developers

书　　名：基于 XML 的 ASP.NET 开发
作　　者：Dan Wahlin
译　　者：王宝良
出 版 者：清华大学出版社（北京清华大学学研大厦.邮编 100084）
http://www.tup.tsinghua.edu.cn
责任编辑：朱起飞
印 刷 者：北京市耀华印刷有限公司
发 行 者：新华书店总店北京科技发行所
开　　本：异 16　　印张：28.625　　字数：589 千字
版　　次：2002 年 7 月第 1 版　　2002 年 7 月第 1 次印刷
印　　数：0001～5000
书　　号：ISBN 7-302-05649-8/TP・3331
定　　价：42.00 元

目 录

第 1 章　XML 和 ASP.NET 开发

除非你是在一个与世隔绝的孤岛上度过了最近几年，否则你一定不会错过围绕XML而兴起的这股热潮。遍布世界的报纸、杂志和Web站点都将XML描绘成许多现存技术问题的解决方案。这种全球范围内的关注将XML推到了舞台的中心，导致许多产品应用都宣称支持XML。然而，XML真的有那么优秀吗？虽然它不会带来世界的和平或降低臭氧的浓度，但它的确是一种强大的技术，当你学会如何使用它以及在何时应用它之后，它可以带给你许多回报。

你可能已经敏锐地意识到，XML提供了一种描述数据的平台无关的方法，可以在许多应用中使用。我们将在后面深入介绍与此相关的许多细节，但在此之前，首先要回答几个问题，这很重要。

首先，XML从哪里来？为什么今天这个面向技术的世界需要XML？更重要的是，作为一个ASP.NET开发人员，如何充分利用XML的强大功能？

本章将给出这些问题的答案，并将把XML与另一种你已经非常熟悉的语言进行比较。后面各章将提供大量的例子，来展现如何在ASP.NET中使用XML来为你的开发项目添加许多有用的功能。在此之前，让我们先来看看XML来自何方。

1.1　XML的祖先——SGML

在标记语言这个世界中，XML是一种相对较新的语言。你可能对XML的近亲语言，即超文本标记语言，或者简称HTML，非常熟悉。标记语言通常用来提供有关数据的元数据。对于XML和HTML，这是通过特殊的标记来完成的，例如HTML中的<font>标记。SQL Server的系统表是元数据的另一个例子。它们提供了有关数据库中自身数据的结构信息。XML具有同样的功能，但它是用来描述基于文本的数据的。

万维网联盟（World Wide Web Consortium，W3C）的XML 1.0规范在1998年2月才推出，而XML的母语言，标准通用标记语言（Standard Generalized Markup Language，SGML）自1986年开始就已经以标准的形式存在了。SGML是一个复杂的语言标准，被许多工业，包

括技术出版工业，广泛采用来标记数据。它使用一种由SGML定义的称为文档类型定义（Document Type Definition，DTD）的语法来指定一个给定的SGML文档的结构。

虽然XML使用起来要容易得多，它的许多功能和规则都是从SGML语言派生而来的。当W3C在1996年第一次开始制定现在称为XML 1.0的规范时，他们从SGML中借用了许多概念，包括DTD。为了使XML更易于使用，以及更加面向Web，SGML中许多复杂的东西都被省略了。如果你有兴趣阅读有关XML和SGML的比较内容，请参看http://www.w3.org/TR/NOTE-sgml-xml-971215。

现在你已经知道了XML起源的一些背景，让我们再来看一看为什么我们需要XML。

1.2 为什么我们需要XML

你可能会发出疑问：现今有许多不同的可用编程语言，为什么我们还需要另一种语言，并且仅仅是用来标记数据？虽然许多语言允许编写不同的接口并与远程计算机进行交互，但很少有这样的语言：它能提供一种平台中立的方法以在完全不同的系统之间共享数据。而允许这种交换可以通过防火墙或其他安全相关的机制，或者为具有不同技术水平的个人提供一种易于使用的格式，这样的语言就更少了。

这就是XML引起人们注意的地方。W3C的XML工作组知道：需要发明一种简单的语言，这种语言易于使用，可以使不同系统之间的数据交换更加高效，并且可以在Web环境下很好地工作。正如前面提到的那样，因为SGML已经在标记数据方面获得了成功，它的许多思想就被用来开发XML子集。这就生成了一种新的语言，这种语言有非常高的灵活性，同时具有非常严谨的结构（取决于个人的需要或者它所运行的系统）。

为了更加深刻地理解我们需要XML的原因，仔细看一下下面用逗号分隔的文件：

```
Elbow Joint, 12930430, 6, 25, 06/28/2000, 1238 Van Buren, B2B
  Supply, 1111236894, Walters
Valve, 39405938, 3, 40, 06/20/2000, 4568 Arizona Ave., A+ Supply, 2221236894,
Tammy PVC, 234954048, 6, 20, 06/14/2000, 49032 S.  51, A+ Supply, 2221236894,
Walters
```

虽然可以对这个文件中的某些数据元素的意义做出一些假设，许多元素都没有为你提供一种方法来理解它们所代表的含义。例如，第1行中的06/28/2000字段代表一个日期吗？如果是这样，什么类型的日期？第4行的234954048字段又表示什么？是一个零件号，一个

客户编号，或者完全无关的什么东西？清单1.1给出了转换成XML的同一个文件。

清单1.1　使用XML标记一个用逗号分隔的文件

```
<?xml version="1.0"?>
<supplies>
   <item supplier="1">
      <description>Elbow Joint</description>
      <partID>12930430</partID>
      <numberInStock>6</numberInStock>
      <numberOnOrder>25</numberOnOrder>
      <deliveryDate>06/28/2000</deliveryDate>
      <supplier>
          <street>1238 Van Buren</street>
          <company>B2B Supply</company>
          <phone>1111236894</phone>
       </supplier>
       <orderedBy>Walters</orderedBy>
   </item>
   <item supplier="2">
      <description>Valve</description>
      <partID>39405938</partID>
      <numberInStock>3</numberInStock>
      <numberOnOrder>40</numberOnOrder>
      <deliveryDate>06/20/2000</deliveryDate>
      <supplier>
         <street>4568 Arizona Ave.</street>
         <company>A+ Supply</company>
         <phone>2221236894</phone>
      </supplier>
      <orderedBy>Tammy</orderedBy>
   </item>
   <item supplier="2">
      <description>PVC</description>
      <partID>234954048</partID>
      <numberInStock>6</numberInStock>
      <numberOnOrder>20</numberOnOrder>
      <deliveryDate>06/14/2000</deliveryDate>
      <supplier>
         <street>49032 S. 51</street>
```

```
37:          <company>A+ Supply</company>
38:          <phone>2221236894</phone>
39:       </supplier>
40:       <orderedBy>Walters</orderedBy>
41:    </item>
42: </supplies>
```

每一个数据项都很容易识别和理解，因为所有的数据都使用了XML标记进行描述。使用这些标记，人和计算机处理数据变得更加容易。

虽然计算机通常能够处理许多数据集中应用程序的自动化，基于一些文件（类似于前面用逗号分隔的平面文件）的程序对数据结构的改动适应性很差。另一方面，XML可以使得应用程序根据描述性标记而不是根据位置来访问数据。这就为提高一个应用程序对数据结构改变的适应性提供了很好的机会。

XML也为在应用程序之间，甚至是应用程序的组件之间能够共享数据提供了一个非常好的机会。通过事先理解一个XML文件的结构，应用程序中的一个组件可以处理包含在XML文件中的数据并执行不同的任务。在某些情况下，数据可能表示有关一个客户或者商业计划书的信息。而在另一些情况下，XML文件可能仅仅表示对象的属性值，ASP.NET组件可以用它来使编程更加简单和高效。

XML还有一个优点，即在集成DTD和schema文件过程中必须事先确定文件采用何种结构。定义这一结构允许具有不同系统的公司顺利地进行数据交换，而不必担心数据最初是存储在何种系统中的。只要双方都知道XML文档的结构会是什么，他们就可以使用XML语法来交换数据。基于XML的应用软件，例如BizTalk Server或者SQL Server 2000，可以使这一过程更加容易，并且当将其与ASP.NET的威力组合后，就可以生成更加高级的应用软件。

最后，正如前面提到的那样，你无需对计算机科学有很深的理解就可以使用XML。看一看清单1.1中的XML代码，你会发现其中并没有使用令人胆怯的字符或者语法。实际上，这个XML文件可以被一个没有任何技术经验的个人阅读并理解。在学习了语法、结构和编程规则之后，就可以创建XML文档了，不管是在简单的还是在复杂的应用中都没问题。

1.3 XML的可扩展性

XML的另一个重要优点是它的可扩展能力。XML的可扩展性允许作为开发人员的你选

择如何命名和构造XML文档中的标记。在HTML中，你只能使用那些以预先定义的方法使用才能生效的特定标记。举例来说，为了在一个HTML文档中放置一幅图像，你必须使用<img>标记。在这个标记之内，你只能设置特定的属性，例如src和vspace属性。XML允许你根据所设计的应用需要如何进行组织来自由地创建标记和属性。因此，如果你愿意使用一个名为<imagePath>的标记来描述一个图像路径，你就可以这样做（只要包括了一个结束</imagePath>标记就行）。

乍看起来，这样做可能会导致彻底的混乱。毕竟，如果每个人都可以使用不遵循任何命名规范的标记来创建XML文档，其他人怎么能够使用这些文档呢？答案是：XML的可扩展性可以通过使用DTD或者schema文档进行控制，后两者定义了一个给定的XML文档必须遵从什么样的结构才是有效的。因此，如果你想使用名为诸如<myTag>或<myData>这样的定制标记创建一个文档来描述数据，你就可以这样做。任何想使用你的XML文档的人也可以使用它，只要你为他们提供一个文档是如何组织以及使用什么样的标记描述数据的定义即可。

要理解XML的可扩展框架的价值，关键是要认识到XML并不指定如何为一个最终用户或者系统显示数据。作为一种标记语言，XML只关心如何描述数据，最终可以使用其他语言来操纵和显示这些数据。为了更加明确这一点，让我们来看一下XML和HTML之间的区别。

1.4　XML和HTML之间的区别

因为XML和HTML都是SGML语言的子集，因此它们在某些方面有类似之处。然而，它们的总体目标是不同的。下面的许多概念将在第2章“在ASP.NET中使用XML基础”中深入讨论。

1.4.1　显示与描述

HTML起初被设计来为远程用户显示基于文本的信息。如果你已经使用Web很长时间，你可能记得Web站点包含标准的灰色背景和黑色文本的那些日子。如果你资格够“老”，你甚至会记得使用一个Lynx浏览器。随着Web应用越来越普及并被看作是为遍布世界的人们显示信息的一种新奇的方法，使用图形、表和其他元素来显示数据的一些其他方法也被

集成到HTML语言中。这些新增技术提供了背景和大量其他的显示能力。你可能记得这样的情形：背景非常鲜艳或者五颜六色，而文本非常刺眼几乎无法阅读。

所有这一切与XML有什么关系？实际上，没什么关系，但是有提到它的原因。主要一点就是HTML是一种显示标记语言，只有在与一个应用，例如一个浏览器，组合起来时才有用。它有许多预定义的标记和属性，可以用来改变字体颜色、使文本变大，或者将那些有趣的粉红色云彩的背景放置到一个页面上。然而，HTML在描述所要显示的数据方面什么都没做。一个包含如下HTML代码的页面没有显示出有关它所包含数据的任何信息：

```
<font color="#02027a" size="3"><b>274943494</b></font>
```

代码中的274943494代表什么？是一个银行帐号，一个零件编号，一个身份证号，或者什么都不是？我们不能通过查看HTML文档来得到这些信息，因为HTML只显示信息，它并不描述信息。

与之相比，XML没有涉及任何显示数据的处理。它的惟一目的是使用标记、属性和其他的项来描述数据。前面HTML代码段中的数据在XML文档中可以用如下的方式编写：

```
<population>274943494</population>
```

现在很容易得知数据究竟表示什么——数据与人口（population）相关。此数据可以使用其他的标记进一步描述如下：

```
<unitedStates>
    <population>274943494</population>
</unitedStates>
```

现在我们知道此数据是与美国相关的一个人口数。

1.4.2 结束标记

HTML提供了许多自由，包括允许标记是开放的，甚至有一些标记是永远都不需要闭合的。例如，在HTML中，一些标记可以是开放的，在浏览器中不会引起任何问题。这里是其中的几个例子：

- <img>
- <hr>
- <p>
-
