XHTML Visual C++ DTDs XSLT Schemas .Net Biztalk SAX

XML-THE MICROSOFT WAY

微软 XML 技术指南

[美]Peter G. Aitken 著
谢君英 译

Addison Wesley
Pearson Education
中国电力出版社
www.infopower.com.cn

XML–THE MICROSOFT WAY

微软 XML 技术指南

[美]Peter G. Aitken 著
谢君英 译

中国电力出版社

XML—the Microsoft Way (ISBN 0-201-74852-5)

图书在版编目（CIP）数据

微软 XML 技术指南 /（美）艾特肯（Aitken,P.G.）著；谢君英译. —北京：中国电力出版社，2003
ISBN 7-5083-1407-7

Ⅰ.微... Ⅱ.①艾... ②谢... Ⅲ.可扩充语言，XML—程序设计 Ⅳ.TP312
中国版本图书馆 CIP 数据核字（2003）第 004554 号

责任编辑：常虹
书　　名：微软XML技术指南
原　　著：（美）Peter G.Aitken
翻　　译：谢君英
出版发行：中国电力出版社
地址：北京市三里河路6号　邮政编码：100044
电话：（010）88515918　传真：（010）88518169
印　　刷：汇鑫印务有限公司印刷
开　　本：787×1092 1/16　**印　　张：**23.5　**字　　数：**524 千字
书　　号：ISBN 7-5083-1407-7
版　　次：2003年8月北京第一版
印　　次：2003年8月第一次印刷
定　　价：45.00 元

前　言

从概念上来说，XML（Extensible Markup Language，可扩展标记语言）是相当简单的。它只是为数据结构化提供一套规则。由于这种底层技术的简单性（或者这只是部分原因），XML 已经变成了一门非常重要的技术，它广泛地用于信息处理的各个领域。因为 XML 与生俱来就是用于各种数据的，所以目前它已经在金融、农业、法律、医药和计算机编程等方面得到了巨大的应用。因为 XML 是公共的、非专有的标准，所以它可以被任何人在任何地方使用。因为它是平台无关的，所以它可以用于 Windows 个人电脑、苹果机、Linux 机器、大型机，以及其他任何你所关注的平台。更为重要的是，XML 允许这些不同类型的计算机轻松地进行数据交换，而这在以前是无法想像的。越来越多的应用程序开发人员发现他们需要编写 XML 应用程序。很多人是因为这种原因才阅读本书的，我想你就是其中之一。

关于本书

本书的读者对象为软件开发人员，他们需要学习 XML 基础知识及其最重要的相关技术，他们也想学习如何在他们的应用中包括 XML 支持。本书是特别为那些使用 Microsoft 开发工具在 Windows 平台上创建应用程序的开发人员设计的。本书大约有一半的篇幅讲解公共 XML 及其相关标准，因此不管对于在何种平台上使用何种开发工具的 XML 开发人员来说，本书都是非常有用的（虽然本书的代码示例使用了 Microsoft 语言）。其他章节特别讲述了通过使用 Microsoft 开发与最终用户工具来应用 XML，这些工具包括：Visual Basic、Visual C++、Internet Explorer、ASP，.NET Framework 等等。

本书假设读者已经有一点或者根本没有使用 XML 及其相关技术的经验，由浅入深地介绍了 XML 相关内容，适合于 XML 初学者及这方面的中级读者。另一方面，编程部分的章节，假设读者已经具备了一段时间特定语言或者技术的使用经验。例如，第 15 章展示了如何通过一个 Visual Basic 程序来操作 XML，但本书不讲授使用 Visual Basic 的基础知识。与 Visual C++和新的.NET 语言 C#相关的章节同样也不包括 Visual C++和 C#方面的基础知识。

Java 是什么

对于许多 XML 程序员来说，Java 是一种可以选择的语言。偏爱这种语言的原因众多，最起码的原因是在使用 XML 时有很多功能强大的 Java 工具可供使用。然而，你将看到在本书中没有涵盖 Java，这是因为本书专门讲述与 XML 相关的 Microsoft 工具，而 Microsoft 的 Java 语言，也就是 Visual J++，已经走到了末路。顺应优胜劣汰的选择结果，Microsoft 公司已经宣布停止更新 Visual J++，因此这个产品的当前版本（Visual J++ 6.0）将是最后的版本了。综上原因，在这本面向 Microsoft 的书中没有安排 Java 的内容，尽管它是一门非常优秀的语言。

然而，虽然这本书就像是 Microsoft 工作间，但对于 Java 爱好者来说，还是有一点吸引力的。最新的 Microsoft 开发工具，统称为 Visual Studio .NET，包括了一种称为 C#（读作 C sharp）的语言。虽然 Microsoft 极力否认，但许多人均将 C#看作是 Microsoft 的 Java 语言的替代者。C#在很多方面类似于 Java，我认为任何 Java 程序员不久就会熟悉 C#。

关于程序清单

贯穿全书，代码示例以两种形式出现：一种是作为代码片段散布在正文中；一种是单独作为经过编号的程序清单出现。大部分情况下，经过编号的程序清单是完整的文件，例如完整的 XML 文件或者是 Visual Basic 源代码文件。少数情况下，程序清单是文件的片段，比如不能独立完成功能但必须是大程序一部分的源代码段。这种情况已经明确地指出来了。

源代码与纠错

本书的所有代码（编号的程序清单）可以从以下网页下载：http://www.pgacon.com/xml_ms.htm。下载的文件是一个 ZIP 压缩包。展开这个 ZIP 文件后，可以看到本书各个章节的文件夹（没有程序清单的章节除外）。这个网页还列出了本书中已经发现并已纠正的所有错误。如果你在阅读本书时发现你认为的错误，请通过 e-mail 告诉我们：peter@pgacon.com。我很高兴能听到来自本书读者的评论和建议。然而，请注意，我只能回复与本书直接相关的问题，不能答复读者可能碰到的任何常见编程问题。

Peter G.Aitken
于 Chapel Hill，北卡罗莱纳州

致　谢

虽然本书只列出了一个作者的名字，但它是很多人共同努力的结果。首先要感谢Addison-Wesley的执行编辑Mary T.O'Brien的指导和对整个写作过程的监督，还有助理编辑Alicia Cary对一些不可避免的细节问题的处理。本书大大受益于Ann Navarro、Robert J.Brunner、Duane Nickull、Robert W.Husted、Dr. Abbas Birjandi和Steve Heckler的整体技术审查。最后，要感谢T.E. "Raj" Rajagopal，他在我意识到自己的C++技术已经变得非常生疏而不足以打动读者时，帮助我写作了第16章。

目　　录

第1章

XML 简 介

本章介绍了 XML 的基础知识及简短历史。通过本章你将了解 XML 是什么、XML 的应用以及它为什么变得如此重要，你还将了解一些 XML 的简短历史及其标准化过程。如果你已经有了使用 XML 的经验，可能已熟悉了这些。但对于 XML 初学者来说，这里所涉及的内容将是学习后续章节的基础。

1.1 XML 是什么？

本质上，XML（Extensible Markup Language，可扩展标记语言）是一门创建结构化数据的技术。术语“结构化”意味着以一种明确的、无二义性的方式标识信息各个单独部分。XML 出现还没有多长时间。XML 1.0 规范于 1998 年发布，但在很短的时间内，XML 就在计算机和数据处理中变得如此重要，这让众多观察家们大跌眼镜。这种重要性在不久的将来会愈发体现出来。那么什么是最重要的呢？包括 3 个方面，这将在下一节中述及。

1.1.1 XML 是一种标记语言

标记语言（Markup Language）是一种存储结构化数据的规范，也就是说，存储数据和关于数据含义的信息。下面是未标记数据的一个简单示例：

```
Jane Smith
123 Oak Street
Durham
NC
27705
```

读取这些数据的人会明白它的含义。当然，很明显 Jane Smith 是一个人名，Durham 是一个城市等等。而对于计算机来说，它不是那么聪明的，它没有一种可靠的方式来知道数据的各部分表示什么意义。标记语言可用于提供这种信息或者结构，如下所示：

```
<person>
  <name>Jane Smith</name>
  <address>123 Oak Street</address>
  <city>Durham</city>
  <state>NC</state>
```


```
  <zipcode>27705</zipcode>
</person>
```

一旦数据如上例所示被标记了，数据的结构就明确了：它由一个称为 person 的元素组成，该元素包括子元素 name、address 等等。程序能够无二义地明确决定信息的各部分表示什么意义。标记语言可采用多种不同的形式。上例中的标记语言（它遵循了 XML 标准）遵守了某种形式的规则。

- 尖括号<...>中的文本是标记语言的标签。尖括号外的文本是数据。
- 数据单元（或元素）的开始用一个起始标签来标记，以识别数据。
- 数据单元的结尾用一个结束标签来标记，结束标签与起始标签相同，只是在结束标签前加了一个斜杠（/）。

在上例中，你可以看到<address>是起始标签，123 Oak Street 是数据，而是</address>是结束标签。

当然，XML 还有更多的规则，这些规则将覆盖全书。就目前来说，你还只需理解 XML 的基本概念，比如，XML 是什么、它能够做什么。虽然标记语言的概念看起来是相当简单的，但它提供了令人叹为观止的强大功能和灵活性。这是因为 XML 文档的结构在处理文档中的数据时没有任何的约束。在学习了更多的 XML 知识后，这一点你会理解得更加透彻。

XML 存储数据的方式没有任何的特别：它将数据保存在标准的文本文件中。可以使用任何的文本编辑器，比如 Windows 操作系统自带的记事本应用程序，来读取和编辑 XML 文件。然而，也要注意，虽然 XML 文件能为人们所阅读，但很少有直接阅读 XML 文件的。大部分情况下，XML 都是通过程序来创建，然后被其他程序所使用，而不会由人来直接查看 XML 文件。

1.1.2 XML 是可扩展的

XML 中的 X 代表“eXtensible”，也就是说，这种语言能够根据需要进行扩展，以满足特定情况的需求。或者也可以从技术上来讲，XML 与其说是一种语言，不如说是一种元语言（metalanguage），即一种定义其他语言的语言。用专业术语来说，这意味着在创建 XML 文档时，不会局限于一套预先定义的标签，你可以为特定应用程序创建所需要的任何标签。XML 标准提供了一套与这些细节相关的规则，比如，如何创建标签、XML 文档如何结构化。但在 XML 框架中，你可以随意地定义和使用标签，以最好地表达数据。如果你的 XML 文档包括一个邮件列表、一个自动部分编目、一个 Web 页，或者还包括加了注释的莎士比亚的 14 行诗，那么你可以用基本上完全自由的方式表现和组织数据。

1.1.3 XML 保持数据存储与数据显示相分离

要使数据有用，通常需要将数据以某种形式显示出来以便于阅读。这里的术语“显示”（display）要从广义上来理解，它包括了各种可能性：

- 在 21 英寸计算机屏幕上的文档。
- 在 3 英寸个人数字助理显示屏上的文档。

- Web 页。
- 转换为语音输出的文本。
- 音乐回放。
- 杂志出版物输出到排字机。

XML 的设计最值得称道的是，XML 文档中的数据存储对于数据的显示绝对没有任何约束。如果数据使用 XML 规范存储，就可以确保对数据的显示已经没有任何限制。XML 数据可以经常被显示，但正如你所知道的，数据的显示与数据的存储是完全分离的。

1.1.4 XML 是公共的和被广泛接受的标准

XML 标准由 W3C（World Wide Web Consortium，万维网联盟）创建。W3C 是一个开放的、公共的组织，其任务是开发 Internet 上的技术和标准。除了 XML 之外，W3C 也是 HTML（Hypertext Markup Language，超文本标记语言）、PNG（Portable Network Graphics，可移植的网络图形；一种 Web 上使用的图形文件格式）、HTTP（HyperText Transfer Protocol，超文本传输协议）标准的支持力量。在本章后面对 W3C 和标准化过程会有更详细的解释。要想了解更多信息，可以参考 W3C 的网站 http://www.w3.org/。

1.2 XML 发展史

为了更好地理解 XML，应该知道关于 XML 开发的一些事情——它的起源、前身以及创建它的动机。在下一节，你将明白标记语言的概念并不是新事物。

1.2.1 早期的标记语言

大约 40 年前，为了方便数据的交换和控制，IBM 公司人员开始寻求文档结构化的标准方法，由此产生了 GML（Generalized Markup Language，通用标记语言），并且在 IBM 公司内部用来创建各种各样的文档。其他机构也相继开发了类似的技术，但都彼此各自所有，互不兼容，当时没有统一的业界标准。

第一个标准化的标记语言同样由 IBM 创建，称为 SGML（Standard Generalized Markup Language，标准通用标记语言），它最初是为使用合法文档而创建的。SGML 随后被拓展成通用的标记标准，且很快得到广泛应用。1986 年，ISO（国际标准化组织）将 SGML 发布为官方标准。SGML 是一项非常强大和灵活的技术，因而不可避免地带来很大的复杂性和处理开销。

1.2.2 HTML

Internet 的出现促进了标记语言的进一步发展。当时，Internet 上出现了各种不同类型的文件（文本、图形等），于是瑞士欧洲粒子物理实验室的工程师 Tim Berners-Lee 认识到，如果用一种可行的方法建立连接，使用户能轻松地在相关文档间移动，那么将大大改进对这些不同类型的文档进行的访问。为了指出文档之间的链接并指出文档在浏览器中的显示方式，需要一种方法来标记这些文档。于是由 SGML 衍生出了 HTML。随着 HTML 语言

的出现，由所有链接文档的 Web 页面组成的万维网（WWW）应运而生了。HTML 是标准的“Web 语言”，我们所见到的任何 Web 页都使用了 HTML 语言。

HTML 取得了巨大的成功，但是伴随着成功的是“成长的烦恼”。随着 Web 的增长，开发人员希望 Web 页面能够包括越来越多的内容——动画、数据访问、交互性等等。然而，HTML 最初只是作为一种超链接和显示标记语言而设计的，因此它显然无法承担此重任。为了从 HTML 制作出预期的效果，许多聪明的开发人员夜以继日，煞费苦心。他们的结果是卓有成效的，这一点每个喜欢网上冲浪的人都有目共睹。

即便如此，仍然可以清楚地预见，在不久的将来，HTML 将不再能够满足 Web 开发人员的要求。他们需要更强大和更灵活的工具。HTML 最严重的缺陷在于它的固定标签集，开发人员可以使用 HTML 中定义的标签集，但仅此而已，因为它没有扩展性。另一方面，SGML 全面支持自定义标签。然而，SGML 的复杂性和处理开销使得它无法适应于普通的 Web 应用。

1.2.3 SGML 用于 Web

1996 年，W3C 着手相关标准的开发工作，要求该标准以某种形式提供适合 Web 使用的、强大而灵活的 SGML。新标准要求中包括了以下 3 条 SGML 最重要的优越性：

- 扩展性（Extensibility）：开发人员能够根据特定应用的需要定义自定义标签。
- 结构性（Structure）：语法遵循定义明确的结构。
- 有效性（Validation）：文档对于数据模型应该是有效的。

W3C 委员会为此付出了差不多两年时间。1998 年 2 月，该新语言标准的第 1 版问世，即可扩展标记语言 1.0 版（Extensible Markup Language version 1.0），它仍然是现在的 XML 规范，用 W3C 的话来说，就是 XML 推荐标准。

1.3 可选的 XML 技术

核心 XML 标准提供了一系列规则，这些规则用来创建标签和属性以提供数据结构化的方法，以及提供 DTD（Document Type Definition，文档类型定义）功能用于约束文档。就自身而言，这个核心标准对于许多用途已经足够了。不过，对于其他与数据相关的应用，核心 XML 标准提供的工具是不够的。因此，为了满足这些需要，围绕着核心 XML 标准发展了以下一些相关的可选技术。

1.3.1 CSS

CSS（Cascading Style Sheet，层叠样式表）是一种用于将显示样式（比如字体、缩进、行间距和颜色等）关联到 XML 文档元素的样式表语言。一种显示程序（例如浏览器）利用 CSS 中的信息来决定如何显示 XML 文档中的数据。使显示规范与数据相分离能够提供巨大的灵活性。这样，当定义不同的样式表时，相同的数据就可以根据特定情况使用不同的显示方式。类似地，如果一个样式表控制了许多文档的显示样式，那么就可以通过修改一个样式表从而轻松地改变所有页面的外观了。

1.3.2 XSL

XSL（Extensible Stylesheet Language，可扩展样式表语言）是一种本质上与 CSS 具有相同用途的样式表语言：用来定义如何显示 XML 文档中的数据。XSL 被公认为比 CSS 具有更强大的功能，所以，相应地学习和使用 XSL 也比较困难。XSL 由两部分组成：XSLT（XSL Transformation，XSL 转换）和 FO（Formatting Object，格式化对象）。XSLT 用来将原始 XML 文档结构转换为当前显示需要的结构。例如，改变数据的顺序和组织，创建新元素，如内容表或索引表。转换也包括为格式化对象添加引用。FO 是指一组指定显示意图的对象，如字体、颜色以及版式等等。已转换文件由一种“格式程序”进行处理，这是一种采用 FO 指令来创建最终输出的程序。一般情况下，FO 用来创建打印输出而 CSS 更多地用于 Web 浏览显示，当然这种区别并不是绝对的。

XSL 并不仅仅适用于数据显示。在没有 FO 的引用下，单用 XSLT 也可以将 XML 文件转换成不同的表现和结构。这种转换的用途非常广泛，它是把 XML 文档转换为 Web 浏览器可显示的 HTML 的最常用方法。

1.3.3 XML Schema

XML Schema 是定义 XML 文档数据模型的标准。所谓“数据模型”是指数据在文档中必须遵循的一套规则。换一种说法，数据模型指定了文档中允许的标签、各标签之间的关系以及每个标签所能包含的数据类型。为了进一步解释，请再次查看本章前面给出的 XML 片段：

```
<person>
  <name>Jane Smith</name>
  <address>123 Oak Street</address>
  <city>Durham</city>
  <state>NC</state>
  <zipcode>27705</zipcode>
</person>
```

上述 XML 文档的数据模型说明如下：

- 标签 person、name、address、city、state 以及 zipcode 都是允许的。
- 标签 name、addres、city、state 以及 zipcode 必须包含在 person 标签内。
- state 标签中的数据必须由两个字母组成。

数据模型说明什么标签是允许的，什么标签是必需的，标签之间的关系以及 XML 文档结构化的其他方面。数据模型是 XML 的重要方面，因为大多数采用 XML 格式数据的应用程序都要求数据以某种特定方式加以组织。这可以通过“有效性”处理来完成。在 XML 文档处理过程中，文档内容按照文件数据模型进行检查。如果文档与模型相符则被认为有效，然后将进行进一步的处理，通常这种数据在类型和结构方面都不会有问题。

XML Schema 标准是一种创建数据模型的方法。而另一种方法 DTD，之所以没有单独提到，如前所述，是因为它本身是核心 XML 1.0 标准的一部分而不是可选标准。事实上，DTD 是创建数据模型的最初方法，并且至今仍被广泛使用。需要说明的是，“模式

（schema）”是一个与 XML 数据模型相关的一般术语，而“XML Schema”是用来创建模式的具体技术。同样，术语“词汇表”有时也与模式一样指同样的事情。因此，模式也可以定义一个 XML 词汇表。

良构的和有效的 XML 文档

符合 XML 语法规则的 XML 文档被认为是良构的。这与文档是有效的是不同的。在良构的文档中没有 DTD 或者模式。一个文档是良构的，但不一定是有效的。但如果一个文档是有效的，就必须是良构的，并且还要符合 DTD 或者模式定义的数据模型。

1.4 XLink 与 XPointer

XLink（XML Linking Language，XML 链接语言）标准定义了创建到外部资源链接的标准方法。由 XLink 定义的链接在概念上类似于在 WWW 上使用的超链接，但更具灵活性。因此，XLink 使用 XML 语法来在资源（文件）之间提供链接，并提供实现双向链接甚至更复杂连接的能力。一个相关的标准 XPointer，在 XML 文档内部提供了链接（从文档的一个部分到另一个部分）。

1.4.1 DOM

DOM（Document Object Model，文档对象模型）是一组标准化的过程，程序和脚本能用该过程来访问和操纵 XML 文档的内容。DOM 提供了对文档中数据以及文档中结构（标签）的访问。DOM 通过解析器变得可用，解析器是一个程序，该程序在处理一个 XML 文档时执行了第一步。你将在本章后面学到关于解析的更多内容。现在知道解析器读取一个 XML 文件并使文件内容对于程序或者脚本可用就够了。程序或脚本使用 DOM 来访问信息。特别情况下，DOM 定义了一组编程过程，统称为 API（Application Programming Interface，应用编程接口），程序或脚本调用该过程来访问 XML 文件内容。如果编写的程序或脚本遵循 DOM API，就可以使用兼容 DOM 的解析器。同样，如果你编写的解析器与 DOM 兼容，它就能为任何使用 DOM API 的程序或脚本所使用。

1.4.2 SAX

SAX（Simple API for XML，XML 简单 API）是一个应用编程接口或 API，其设计初衷与 DOM 相同——也就是说，提供到 XML 文档内容的可编程接口。虽 DOM 与 SAX 服务于相同的基本目标，但它们也有很大程度的不同。这两种技术的详细内容将在本书的后面章节介绍。现在只要明白 SAX 不如 DOM 功能强大就够了（虽然 SAX 对于许多应用来说已经是足够完美了）。不过 SAX 在内存和处理能力方面有较少的开销。SAX 和 DOM 可以同时使用，一个给定的应用程序可以使用这两种 API，也可以选择其中任何一种作为特定任务最合适的来使用。

1.4.3 XML 命名空间

XML 命名空间标准允许复合 XML 文档的创建。复合文档（compound document）是一种由两个或多个彼此基于不同词汇表（不同的 DTD 或模式）的片段组成的文档。由于带有两个或多个词汇表，所以无法保证相同的标签名不被两个不同用途的词汇表使用。为了阻止这种可能的混淆，命名空间标准提供了一种方法来明确地识别哪种词汇表对于哪个文档和文档的哪个部分是可用的。

1.5 XML 标准处理

标准一词有两种不同但相关的含义。它通常的意义是一个被广泛接受的认可：关于某事意味着什么或者某事应该怎样算是完成，而不隐含任何官方的状态或批准。这里我们使用的是这个含义。然而，另一种情况下，标准是一个国际上经过官方正式协商通过的规范或文档，并由某个组织来发布。在现代生活的几乎所有领域夸大标准的重要性是不可能的。许多通常的事情都得到了我们的认可，比如，米的长度、夸脱的容量、秒的时间长度、这些都是建立在标准之上的。

标准在计算机和数据处理领域也同样重要。举一个例子，考虑最重要和最广泛使用的标准之一 ASCII（American Standard Code for Information Interchange，美国信息交换标准码），有时也称为 ANSI 标准，由美国国家标准学会发布。由于计算机使用数字，所以通过计算机内的数字表示字母和其他符号是必需的。ASCII 标准指明了字母、标点符号、数字和符号的编码，这些都用于文本文档中。保存为磁盘文件的字母可能是值 72 101 108 108 111，但任何遵循 ASCII 标准的程序都会将这个值解释为“Hello”。ASCII 标准是普遍接受的，这意味着文本文档可以在不同程序和不同计算机之间自由交换。你可以想像一下，如果不同程序和操作系统为不同字母和符号使用不同的编码将带来什么问题。

1.5.1 W3C

世界上有许多组织处理标准（再次强调，这是广义上的“标准”），但与本书主题相关的标准组织是 W3C。该组织由 WWW 的发明者 Tim Berners-Lee 于 1994 年在麻省理工学院创建。现在，W3C 已经由 3 个学院平等地控制：麻省理工学院、法国国家计算研究中心（French National Computing Research Center）、日本庆应义塾大学（Keio University）。

Berners-Lee 和其他人认识到 Internet 和 Web 的未来发展和有用性在很大程度上依赖于适当标准技术的可用性。W3C 最重要的任务就是设计这些技术，并发布这些技术的公共规范。

W3C 目前由超过 400 个成员组成。这些成员不是单个的人，而是一些组织，比如软件与硬件厂商、高等院校的协会、公司、政府部门以及内容提供商。任何对 W3C 的活动感兴趣的组织都可以加入。