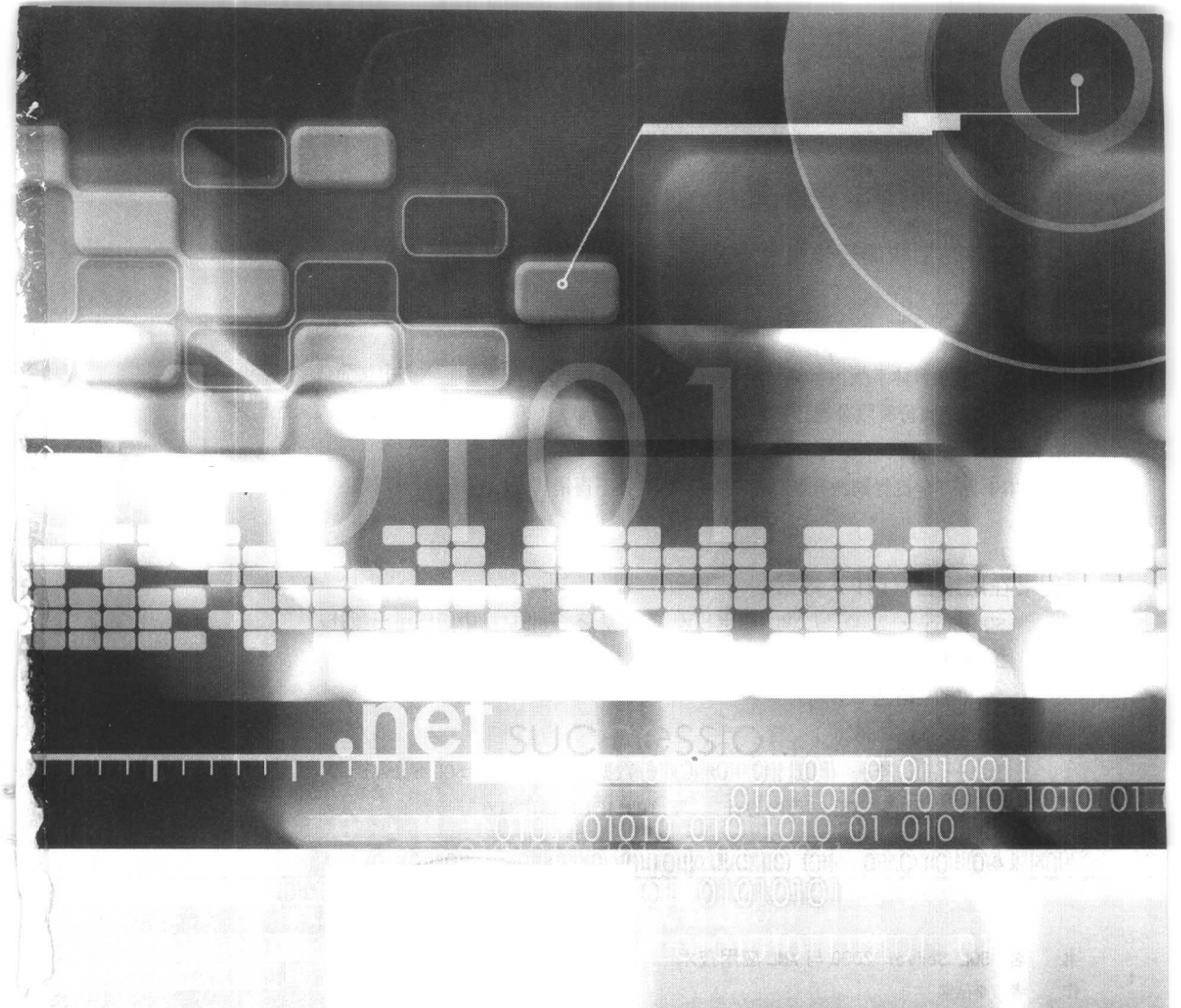




中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



SQL Server 2000 与 XML 应用技术

李昆堂 著

中国铁道出版社

2003·北京

(京)新登字 063 号

北京市版权局著作权合同登记号: 01-2002-3897 号

版 权 声 明

本书中文繁体字版由台湾学贯行销股份有限公司出版(2002)。本书中文简体字版经台湾学贯行销股份有限公司授权由中国铁道出版社出版(2002)。任何单位或个人未经出版者书面允许不得以任何手段复制或抄袭本书内容。

本书贴有学贯行销激光防伪标签,无标签者不得销售。版权所有,侵权必究。

图书在版编目(CIP)数据

SQL Server 2000 与 XML 应用技术/李昆堂著. —北京:中国铁道出版社,
2002. 11

(.NET 战略丛书系列)

ISBN 7-113-05030-1

I. S… II. 李… III. ①关系数据库—数据库管理系统, SQL Server

②可扩展语言, XML—程序设计 IV. TP311.138②TP312

中国版本图书馆 CIP 数据核字(2002)第 094912 号

书 名: SQL Server 2000 与 XML 应用技术

作 者: 李昆堂

出版发行: 中国铁道出版社(100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 郭毅鹏

责任编辑: 苏 茜 黄园园

封面设计: 孙天昭

印 刷: 北京兴顺印刷厂

开 本: 787×1092 1/18 印张: 20.75 字数: 492 千

版 本: 2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷

印 数: 1~4000 册

书 号: ISBN 7-113-05030-1/TP·829

定 价: 35.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

“Visual Studio .NET 战略丛书” 序

两年前微软宣布.NET 战略到现在,.NET 已经从战略变成现实。.NET 给计算机业带来了全新的、快速而敏捷的计算能力,也给软件开发人员提供了支持未来计算的高效 Web 服务开发工具,把计算机编程简化得更容易,为全球软件开发人员提供功能更加强大、界面更加友好的开发工具。Visual Studio .NET 中文版已经于 2002 年 3 月在中国推出。

Visual Studio .NET 是一个功能强大、高效并且可扩展的编程环境。有许多激动人心的新功能,提供统一的集成开发环境,支持在同样的开发环境里用 Visual Basic、Visual C++、Visual C#、Visual J# 和其他数十种编程语言,编写、调试和部署各种应用程序。它充分展现了应用程序开发的潜能,并提供了生成应用程序所需的工具和技术。这些应用程序将给当今的企业、机构提供强大的支持,为开发下一代以 XML Web 服务为中心的应用程序而设计,并推动下一代基于 XML Web 服务软件的发展,是有史以来功能最强大、最受欢迎的软件开发工具。

有了 Visual Studio .NET,那些对全世界数百万的专业和业余程序员来说曾一度极端复杂、费时费力,甚至让人望而生畏的编程任务现在已不再难以跨越。为了使开发人员能够运用.NET 技能和知识来迎接编程挑战,铁道出版社从中国台湾地区引进了一套“Visual Studio .NET 战略丛书”,本套丛书精选台湾各家 IT 出版社的.NET 书籍,全部为台湾地区精品上榜图书,充分让读者感受.NET 各个新功能和特性,是一套全面性解说的经典之作。

本套丛书的特色:

- 以面向对象的观点来论述,微软权威专家倾情力著
- 丰富完整实用的例子展现开发的方方面面,STEP BY STEP 引领读者步步提高
- 理论讲解与范例相结合,结合实际项目开发中积累的经验给出最实用的建议
- 深入浅出介绍最新 Web 开发工具,诠释构建 Web 应用程序的新思路

经过紧张的策划、设计和制作,由中国铁道出版社计算机图书中心审选,李之明、杨眉、谭盈、李伟、王君、李赋、许莉、郭少军、陈贤淑、廖康良及孟丽花等同志完成了本套书的整稿和编排工作,丛书终于同读者见面了。严谨、求实、高质量、高品味,始终是我们追求的目标。尽管倾注心血,全力以赴,但错误和不足之处在所难免,恳请读者不吝赐教与指正。我们定会努力工作,不断改进,创作出更多高质量计算机精品图书,以飨读者。

编者

2002 年 12 月

Visual Studio .net 战略丛书

Visual Basic .net

ASP .net

1



Visual Basic.net 完全探索

作者：恒逸资讯 罗慧真

☐ 初级 ☐ 进阶 ☐ 高级

适用读者：有兴趣了解 Visual Basic.net与.net Framework技术的程序设计人员。

2



Visual Basic.net 实战演练

作者：恒逸资讯 胡百敬、罗慧真

☐ 初级 ☐ 进阶 ☐ 高级

适用读者：针对Visual Basic专业开发者，说明Visual Basic.net的进阶议题，包括：.net程序编写风格、面向对象、数据访问技术，以及与旧版本Visual Basic的整合等。

3



Visual Basic.net 窗口应用程序设计

作者：恒逸资讯 罗慧真

☐ 初级 ☐ 进阶 ☐ 高级

适用读者：有兴趣在.net Framework平台开发窗口应用程序者，主题包括：窗口与控件、ADO.net、类、自定义控件，以及组件安装与部署等。

4



Visual Basic.net 彻底研究

作者：恒逸资讯 胡百敬

☐ 初级 ☐ 进阶 ☐ 高级

预计出版时间：2003年

1



ASP.net交互式Web数据库程序设计

作者：廖信彦

☐ 初级 ☐ 进阶 ☐ 高级

适用读者：具备 ASP 开发经验、程序设计基础，有Web数据库基本概念者为佳。

2



ASP.net 与数据库程序设计

作者：桂思强

☐ 初级 ☐ 进阶 ☐ 高级

适用读者：具备网站与数据库开发经验，曾使用至少一种开发语言，如 Visual Basic 或 C 语言等。

目 录

第 1 章 XML 概述	1
1-1 使用 XML 的意义	2
1-2 什么是 XML	2
1-3 XML 基本语句	5
1-4 XML 结构与组成	5
1-5 产生格式正确的 (Well-Formed) XML 文件	8
1-5-1 具有有效合法的 (Valid) XML 文件	9
1-5-2 常用工具	9
1-6 命名空间 (Namespace)	15
第 2 章 DTD 文件类型声明	19
2-1 文件类型声明	20
2-2 如何引用 DTD	20
2-3 DTD 使用须知	22
2-3-1 声明元素	22
2-3-2 元素的数据类型	24
2-3-3 元素的属性声明	26
第 3 章 XDR Schema 声明	43
3-1 XML Schemas 与 DTD	44
3-1-1 引用 XDR Schemas	44
3-2 定义元素	46
3-2-1 ElementType 元素	46
3-2-2 element 元素	49
3-3 定义属性	51
3-3-1 AttributeType 元素	51
3-3-2 attribute 元素	54
3-4 description 元素	55
3-5 datatype 元素	56

SQL Server 2000 与XML

应用技术

3-6 group 元素	56
第 4 章 Xpath	59
4-1 XPath 简介	60
4-2 位置路径 (Location Path)	61
4-3 位置步 (Location Steps)	63
4-3-1 坐标轴 (Axis)	63
4-3-2 节点测试 (Node Tests)	66
4-3-3 选取谓词 (Predicates)	67
4-4 函数与运算符	69
第 5 章 CSS 与 XSLT	71
5-1 使用 CSS 来显示 XML	72
5-2 认识 XSLT	74
5-2-1 XSLT 样式表结构	74
5-2-2 定义模板规则	75
5-2-3 应用 XSL Style Sheet	76
5-3 模板规则	76
5-3-1 定义多个模板规则	78
5-3-2 排序 (Sort)	79
5-3-3 指定输出格式	80
5-4 XSL 文件的合并使用	82
5-4-1 include 功能	82
5-4-2 import 与 include 使用规则	83
5-5 XSLT 的程序设计	84
5-5-1 Template 的名称使用	84
5-5-2 变量与参数	84
5-5-3 循环与流程控制	87
第 6 章 如何提取 XML 数据	89
6-1 提取 XML 格式数据	90
6-2 使用 FOR XML 子句	90
6-3 使用 RAW 模式	91
6-4 使用 AUTO 模式	93
6-5 控制 XML 输出	94
6-6 指定 XMLDATA 选项	99

6-7 提取二进制数据	99
6-8 自定义 XML 属性格式	101
6-9 使用 EXPLICIT 模式	102
6-10 在多数数据表中使用 EXPLICIT 模式	103
第 7 章 使用 Tansact-SQL 维护 XML 数据	105
7-1 产生行集 (Rowset) 数据行对象	106
7-2 产生内部描述树	107
7-3 由内部描述树中取得行集	108
7-4 新建 XML 数据到数据表中	109
7-5 使用 Flags 指定属性或元素对应	110
7-6 使用现存的数据表结构	111
7-7 指定字段对应到不同节点	112
7-8 使用字段节点对应代替 Flag 组合	113
第 8 章 使用 HTTP 访问 XML 数据	115
8-1 设置 HTTP 访问 SQL 数据库	116
8-1-1 测试 HTTP 链接 SQL Server	123
8-1-2 URL 字符编码	124
8-2 在 Server 端使用 XML Templates	125
8-2-1 使用 XML 模板文件	125
8-2-2 在模板文件中使用参数	126
8-2-3 在模板文件中指定样式表	127
8-3 由客户端传送 Template	128
8-3-1 使用 HTML 表单传送模板	129
8-3-2 使用 XML DOM 对象直接传送模板	131
8-4 发布数据库对象	132
8-4-1 使用 HTTP 提取数据库对象	134
第 9 章 使用 ADO 访问 XML 数据	137
9-1 使用 ADO 提取 XML 数据	138
9-1-1 以 XML 方式显示 Recordset	138
9-1-2 使用 Command 对象进行 XML 查询	139
9-2 在 ASP-Based Web 上使用 ADO 与 XML	141
9-2-1 应用 XSL Style Sheet	141
9-2-2 指定输出编码内容	142

SQL Server 2000 与XML

应用技术

9-2-3 使用 ASP 与 ADO 在 Web 上显示数据.....	142
第 10 章 对应 XDR 结构描述文件	145
10-1 对应 Schemas 的使用	146
10-1-1 定义一个默认 Table-to-Schema 的对应.....	146
10-1-2 在 Schemas 中加上注释	147
10-2 使用 Schemas 提取数据	148
10-2-1 使用 XPath 语句提取特定数据	148
10-2-2 在 Command 对象中指定使用 Schame	149
10-2-3 在模板中使用 Scheme 查询.....	151
10-2-4 使用 Schema 虚拟名称查询.....	152
10-2-5 控制 Schema cache	154
10-3 多数据表对应	154
10-3-1 对应具有关联的数据表	155
10-3-2 控制 XML 文件层次结构问题.....	157
10-3-3 筛选数据	158
10-4 现存 Schema 的注释	160
10-4-1 使用 Constants.....	160
10-4-2 排除不必要的字段	162
10-4-3 对应 ID、IDREF 以及 IDREFS 属性	163
10-4-4 指定数据类型	164
第 11 章 使用 Updategram XML 文件修改数据	167
11-1 使用 Updategram 修改数据.....	168
11-2 对应 SQL 的数据到 Updategram 中.....	168
11-3 使用 Updategram 新建数据.....	170
11-4 使用 Updategram 删除数据.....	172
11-5 使用 Updategram 更新数据.....	173
11-6 在 Updategram 中使用参数.....	174
11-7 在 Web 上应用 Updategram	175
第 12 章 使用 SQLBulkLoad 对象	179
12-1 使用 Bulk Load Facility.....	180
12-2 使用 SQLBulkLoad 对象.....	180
12-3 使用 SQLBulkLoad 新建数据表.....	185
12-4 处理 SQLBulkLoad 错误事件.....	187

第 13 章 XML 实用范例—供货整合	191
13-1 范例程序安装说明	192
13-2 范例程序应用说明	195
13-3 范例程序模拟状况	195
13-4 范例程序操作说明	197
13-4-1 在线产品浏览	197
13-4-2 会员注册与登录	203
13-4-3 购买产品与购物袋处理	206
13-4-4 结账处理	210
13-4-5 订单出货处理过程	214
13-4-6 快递公司快递处理	220
第 14 章 使用 XML Schemas	223
14-1 XML Schemas 数据类型	224
14-2 如何引用 XML Schemas	226
14-3 XML Schema 文件结构	229
14-4 默认数据类型	232
14-5 自定数据类型	237
14-6 数据类型分类与组成	241
14-6-1 原生 (Primitive) 与派生 (Derived) 以及 基础 (Base) 数据类型	242
14-6-2 原子型 (Atomic)、列表型 (List) 与 联合型 (Union) 数据类型	243
14-6-3 匿名型 (Anonymous) 与署名型 (Named) 数据类型	245
14-6-4 数据类型的组成结构	245
14-6-5 SimpleType 与 ComplexType 数据类型	247
14-7 Schema 组件的领域性与 Namespace	249
14-7-1 全域性与区域性的对象差异性	250
14-7-2 TargetNamespace 与 default namespace 的意义	250
14-7-3 Qualified 与 Unqualified 的差异	251
14-8 XML Schema 元素使用说明	253
14-8-1 all 组元素	254
14-8-2 any 元素	255
14-8-3 anyAttribute 元素	258

SQL Server 2000 与XML

应用技术

14-8-4	anotation 与 appinfo、documentation 元素	261
14-8-5	attribute 元素	262
14-8-6	attributeGroup 元素	265
14-8-7	choice 元素	265
14-8-8	simpleContent、complexConten 与 extensions 元素	266
14-8-9	field 与 selector 元素	268
14-8-10	key、unique 与 keyref 元素	268
14-8-11	list 元素	278
14-8-12	group 元素	280
14-8-13	restriction 元素	283
14-8-14	sequence 元素	285
14-8-15	import 与 include 元素	286
14-8-16	union 元素	291
14-8-17	redefine 元素	293
14-8-18	element 元素	297
14-8-19	schema 元素	305
14-8-20	simpleType 与 complexType 元素	306
14-8-21	notation 元素	307
14-9	pattern 与 XML Schema 正规表达式	308
附录 A	安装 SQLXML 2.0	313
附录 B	安装 HTTPSQL 练习工具	317
附录 C	安装 XSL 练习工具	319

1

XML 概述

1-1 使用 XML 的意义

因特网（Internet）技术让我们可以与任何地方的任何人通讯。HTML、GIF 以及 Client Side Script 等目前众多标准可在页面建立后，让许多用户无限次显示，并且提供了可视化显示与用户界面。

虽然可视化和用户界面标准，是一个不可或缺的层次，但对要显示且同时要管理数据而言，功能仍显不足。在今天，Internet 只是一个访问文字及图片的媒体，并没有标准可以用于智能型搜索、数据交换、调整式显示以及个性化。

Internet 不只是设置信息访问、显示的标准。它必须是更进一步设计为解释信息标准，一个共同的数据表示方式，让软件得以更快地从众多的文字中搜索、移动、显示和操作其中的信息。但是 HTML 并不具备这样的功能，因为它只是陈述网页的外观格式，而不表示数据。虽然 HTML 可以提供丰富的显示机制，却不能提供一个管理数据的标准方式。

基于标准的数据表示方式，让网络间传送结构化数据，变得和今日在网络上传送 HTML 页面一样简单，它将会启发更多的新用法。这个数据标准就是 XML 以及 XML 的扩充功能。

1-2 什么是 XML

XML 是一种媒体标记语言（Meta-markup Language），提供描述结构化数据的格式。这将有助于文件内容的精确表述，并让跨平台的搜索结果符合文件原意。此外，XML 也促成新一代的网络数据搜索与操作的应用程序的诞生。

■ 数据以结构化表示

XML 提供数据的结构化表示方式，可轻易地开发并广泛地运用。XML 是 SGML 的子集合，并针对网络特性做优化处理，它由 World Wide Web Consortium（W3C）所定义，确保结构化数据的格式一致，并且不受应用程序或厂商影响，促成新一代业务与电子商务的 Web 应用程序。

■ XML 文件

XML 是设计用来存储和传输数据，以文字为基础的格式，且许多方面与 HTML 相似。XML 文件是由 XML 元素所组成，其中每一个元素皆含有起始标志（Start Tag）和结束标志（End Tag），位于两个标志（Tag）间的信息即为内容。与 HTML 一样，XML 文件包含使用卷标注释的文字，但与 HTML 不同的是 XML 允许无限制的标签（Tag）组，每一组标签所代表的不是元素的外观，而是元素的涵意。例如，XML 的

第 1 章 XML 概述

某元素，可能被标注为价格、订单数量或订购者，要使用何种数据，以及哪一种标签名称最适合，完全都由文件的编辑者决定。

一个简单的 XML 文件

```
<xml version="1.0" encoding="gb2312"?>
<订单明细>
  <订购日期>10/20/2000</订购日期>
  <订购时间>08:00</订购时间>
  <订购者>
    <姓名>李汤姆</姓名>
    <地址>台北市金华街100号</地址>
  </订购者>
  <订单数量 单位="箱">100</订单数量>
  <价格 单位="新台币">10</价格>
</订单明细>
```

■ 数据显示和处理分离

XML 的功能强大之处，在于它保持用户界面与结构化数据的独立。HTML 叙述如何在浏览器中显示资料，XML 则定义其内容。在 HTML 中，您使用标签来告诉浏览器将资料显示成粗体或字型颜色；而使用 XML 只要使用标注来描述数据，如客户名称、价格和数量。在 XML 中，您可以使用样式表 XSL 和 CSS 在浏览器中显示数据。XML 将数据的显示及处理分隔开来，只需套用不同的样式表和应用程序，便可显示并处理数据，更可将数据由显示中分离，使得不同来源数据可以完全集成。客户信息、采购单、订货单、应付款项和其他不同来源的数据，可在中间层转换成 XML，让数据可以在线交换，就像现在用 HTML 页面显示资料一样简单。用 XML 编写的数，在集成后可以由网络传递到处理平台。存储到数据库或已有的文件信息中时无需重整，因为使用 HTTP 来传递 XML 数据时，无需额外修改，便可使用此项功能。

■ XML 的自我表述能力

XML 采用文件类型定义 (Document Type Definitions, DTD) 附在文件中。其定义包含了有哪些组件，及组件之间的结构关系等文件规则。当接收的应用程序不具有输入数据的说明时，DTD 可以协助标识数据。经 DTD 验证的数据称为“有效合法的 (Valid)”XML 文件。DTD 所定义的规则，确认数据的结构是否正确。数据传送时，如不具有 DTD 的，称为“格式正确的 (Well-Formed)”XML 文件。不论是使用“有效合法的”或“格式正确的”的 XML，由 XML 编写的数均可自我表述，因为用来描述的标志，已经被包含在数据中。



SQL Server 2000 与XML

应用技术

一份包含 DTD 规则的 XML 文件

```
<?xml version="1.0"?>
<!DOCTYPE EMAIL [
  <!ELEMENT EMAIL (TO , FROM , CC , SUBJECT , BODY)>
  <!ELEMENT TO (#PCDATA)>
  <!ELEMENT FROM (#PCDATA)>
  <!ELEMENT CC (#PCDATA)>
  <!ELEMENT SUBJECT (#PCDATA)>
  <!ELEMENT BODY (#PCDATA)>
]>
<EMAIL>
  <TO>tomlee@mail2000.com.tw</TO>
  <FROM>Bill@msn.com</FROM>
  <CC>Philip@msn.com</CC>
  <SUBJECT>Happy new year</SUBJECT>
  <BODY>Happy new year ! </BODY>
</EMAIL>
```

XML-Schemas 是 XML 文件规则的正式描述, 它指定文件中所允许的元素, 及其可能的组合, 且提供和 DTD 相同的功能。不过, 因为这些 Schemas 语言具有可扩充性, 开发人员可以使用额外信息, 如数据类型、继承性以及显示规则来加强它, 也使得这些新的 Schemas 语言, 远比 DTD 的功能更为强大。通过 Schemas 的使用, 作者可以精确地定义文件中所许可的组件名称, 以及在每一个组件中, 允许何种子组件、属性和关系。更可以从其他架构导入 Schemas 的片段, 然后通过继承以扩充其类型, 也可允许组件间拥有复杂的关系, 但仍维持语汇树状目录结构的简单性。

范例 1-1: 一份 XML Schemas 文件

```
<?xml version="1.0"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data" xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="title" dt:type="string"/>
  <ElementType name="author" dt:type="string"/>
  <ElementType name="price" dt:type="float"/>
  <ElementType name="book" content="eltOnly">
    <element type="title" maxOccurs="1"/>
    <element type="author" minOccurs="1" maxOccurs="1"/>
    <element type="price" maxOccurs="1"/>
  </ElementType>
</Schema>
```

1-3 XML 基本语句

XML 文件主要包含标志 (Tag) 与文字 (Text)。标志用来定义数据成为一个元素 (Elements)，而文字 (Text) 用来展现实质的属性。

■ 产生一个元素 (Elements)

我们可使用以下所示的语句，产生一个元素。

```
<MyName> Michael </MyName>
```

- “<MyName>”：称为起始标志 (Start Tag)
- “</MyName>”：称为结束标志 (End Tag)
- “Michael”：为元素属性 (Element Content)

MyName 为元素名称，而“<起始标志> 元素属性 <结束标志>”这样的组合称为一个元素 (Element)。

■ 指定元素属性 (Attribute)

我们除了可以指定每一个元素的属性之外，还可以指定此元素的相关属性，例如：

```
<MyCar color="Yellow" cc="2000" speed="200">Volkswagen </MyCar>
```

在这个例子中我们指定了三个属性，属性名称分别为 color、cc 与 speed，属性值分别为 Yellow、2000 与 200。

■ 层次化 (Nested) 结构

XML 文件为层次化 (Nested) 结构，一个元素可包含一或多个子元素，例如：

```
<books>
<book id="0001">
<bookname>XML与SQL Server 2000的应用技术(上)</bookname>
</book>
<book id="0002">
<bookname>XML与SQL Server 2000的应用技术(下)</bookname>
</book>
</books>
```

books 元素包含了 book 与 bookname 两个子元素。

1-4 XML 结构与组成

XML 文件包含两大部分：文件标头 (Prolog) 与文件元素 (Document Element)。

SQL Server 2000 与XML

应用技术

文件标头 Prolog	<pre><?xml version="1.0" encoding="gb2312"?> <?xml-stylesheet Type="text/xsl" href="First.xsl"?> <!DOCTYPE booklist SYSTEM "booklist.dtd"> <!--booklist last updated 2000-11-01--></pre>
文件元素 Document Element	<pre><booklist> <book isbn="72-80081-082"> <title>The Gourmet Microwave</title> <publisher>New Moon Books</publisher> <price>9.95</price> <author>Charlotte M. Cooper</author> <others/> </book> </booklist></pre>

■ 文件标头（Prolog）

在 XML 文件中，第一个结构性的元素是文件标头（Prolog），文件标头由两个基本组件构成：XML 声明与文件类型声明，它们都是选择性的组件。

■ XML 声明

XML 声明中定义了文件遵守的 XML 规格版本。这行文字描述必须使用小写字母，在 XML 声明中同时也包含编码声明和单一文件的声明。

- **编码声明：**用来标识字符编码系统。如上例中我们声明以 gb2312 中文编码方式，不同的编码系统会对应到不同的字符格式或语言。例如默认的 UTF-8 包含大部分英文的字符，所以 XML 解析器必须要支持 Unicode 编码标准，才能够支持大部分的语言。
- **单一文件声明：**用来标识 XML 文件是否有使用外部标志的声明。属性值使用 yes 会指示解析器所有关于文件的信息都已经包含在文件中，这表示文件中没有指定外部的实体（Entity），也没有使用外部的结构（External Schema）来指定文件中的默认属性值；属性值使用 no 会指示应用程序需要取得文件以外的信息来完成文件解析。如下所示：

```
<?xml version="1.0" encoding="Big5" standalone="no"?>
```

■ 注释（Comment）

可存在 XML 文件任何地方，例如上例中的：

```
<!--booklist last updated 2000-11-01-->
```