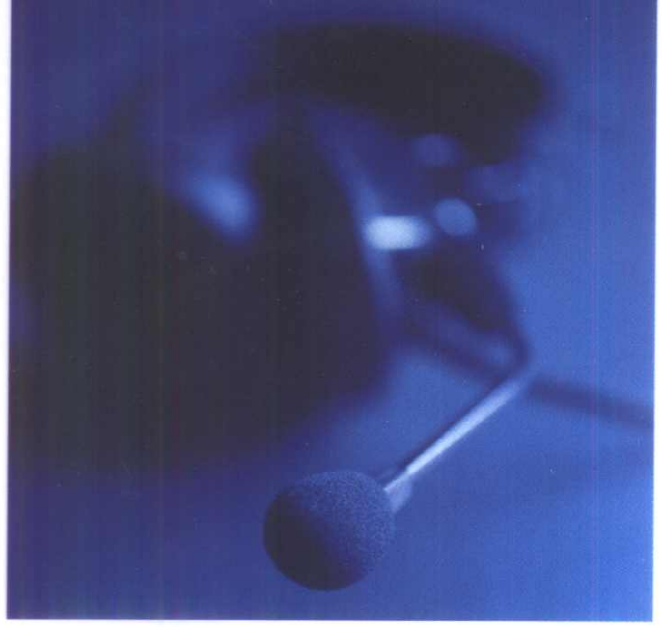

Addison Wesley

XML精要

快速参考手册

—— XML、XPath、XSLT、XML Schema、SOAP

“如果您要开发与XML相关的应用程序，请将本书放进您的工具箱里，它非常有用。”

—— 微软公司.NET XML Framework项目经理Mark Fussell

[美] Aaron Skonnard
Martin Gudgin 著

牛韬 英宇 译

2XM

人民邮电出版社

XML 高级编程快速参考手册

——XML、XPath、XSLT、XML Schema、SOAP

[美] Aaron Skonnard，Martin Gudgin 著
牛韬 英宇 译

人民邮电出版社

图书在版编目(CIP)数据

XML 精要快速参考手册：XML、XPath、XSLT、XML Schema、SOAP /（美）斯科那德（Skonnard，A.），（美）古德金（Gudgin，M.）著；牛韬，英宇译. —北京：人民邮电出版社，2002.10

ISBN 7-115-10571-5

Ⅰ. X… Ⅱ. ①斯…②古…③牛…④英… Ⅲ. 可扩充语言，XML—程序设计—手册

Ⅳ. TP312-62

中国版本图书馆 CIP 数据核字(2002)第 062665 号

XML 精要快速参考手册

——XML、XPath、XSLT、XML Schema、SOAP

◆ 著 [美] Aaron Skonnard Martin Gudgin

译 牛 韬 英 宇

责任编辑 陈 昇

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 http://www. ptpress.com.cn

读者热线 010-67132705

北京汉魂图文设计有限公司制作

北京顺义振华印刷厂印刷

新华书店总店北京发行所经销

◆ 开本：787×1092 1/16

印张：18

字数：426 千字 2002 年 10 月第 1 版

印数：1-4 000 册 2002 年 10 月北京第 1 次印刷

著作权合同登记 图字：01－2002－3736 号

ISBN 7-115-10571-5/TP・3045

定价：33.00 元

前　言

对于任何人，本书都和当前主流 XML 技术一起运用。本书设计成回答许多公共的 XML 相关技术问题的方便而完整的快速参考手册。

本书完全覆盖每一个主题，拥有大量有用示例的传统软件包参考设计。每一章都有概要介绍，并伴随着详细的参考信息。

本书采用的方法是假设读者已经基本理解书中给定的每一个主题。

目　录

第 1 章　XML 1.0 与命名空间

XML 1.0 和 XML 中的命名空间为结构化数据和将标记应用于文档提供了基于标记的语法。符合 XML 1.0 和 XML 规范的命名空间的文档，可能由多种语法结构组成，比如元素、命名空间声明、属性、处理指令、注释和文本。本章提供了 XML 中的每种结构化元素及其各自语法。

1.1　元　　素

```
<tagname></tagname>
<tagname/>
<tagname>children</tagname>
```

通常，元素典型地组成了 XML 文档中的大部分内容。每个 XML 文档都准确地有一个顶层的元素，也就是我们所知的***文档元素***（***document element***）。元素有名字，也可能有后代。这些后代可能本身就是元素或者是处理指令、注释、字符数据（CDATA）段或者字符。元素的后代是有序的。元素也用属性来描述。元素的属性是无序的。元素可能有与之关联的命名空间声明。元素的命名空间声明是无序的。

元素由一对标记串行化而成：起始标记和终止标记。起始标记的语法是一个小于号（<），后面紧跟元素名称，也就是紧跟着我们所说的***标记名***（***tagname***），然后跟着一个大于号（>）。终止标记的语法是字符序列</，后面紧跟着标记名，然后跟着大于号（>）。元素的后代则串行化在父元素的起始标记和终止标记之间。在元素没有后代的情况下，元素称为***空元素***（***empty***）。有一种速记语法可以用于表示空元素，就是由一个小于号紧跟着标记名和字符序列/>组成。

XML 并没有定义任何元素名称，它允许 XML 的设计者来选择要使用的名称。XML 中的元素名称是区分大小写的，它必须开始于字母或者下划线（_）。然后可以紧跟任意长度的字母、数字、句点（.）、连字符（-）、下划线（_）和冒号。但是，由于冒号在 XML 中还用于命名空间的语法中，因此它们不应该用于元素名称中，除非在其规范中做了描述（参看第 1.2 节）。以字符序列 xml 开始的元素名称，或者关于它的任何调整，在 XML 的规范里都被保留以待将来使用。

例子

带有后代的元素

```
<Person>
  <name>Martin</name>
  <age>33</age>
</Person>
```

元素的标记名是 **Person**。这个元素有两个后代，标记名分别叫作 name 和 age。这两个后代元素都带有文本内容。

空元素

```
<Paid></Paid>
```

标记名为 **Paid** 的空元素。

速记的空元素

```
<Paid/>
```

使用速记语法的标记名叫作 **Paid** 的空元素。

1.2 元素、命名空间和命名空间声明

```
<prefix:localname xmlns:prefix='namespace URI'/>
<prefix:localname xmlns:prefix='namespace URI'></
    prefix:localname/>
<prefix:localname xmlns:prefix='namespace URI'>children</
    prefix:localname/>
```

因为 XML 允许设计者自己选择标记名，所以就可能有两个或者更多的设计者为某些元素或者所有元素选择相同的标记名。XML 命名空间提供了一种方法来明确地区分这些 XML 元素：有相同的本地名称，但实际上来自不同的词汇表。这就要将元素与命名空间进行关联来实现。命名空间担当着与之关联的所有元素的作用域。命名空间本身也有名称。命名空间的名称是统一资源标记符（Uniform Resource Identifier，URI）。这种 URI 是一个惟一的字符串，不能够废除。命名空间的名称和元素的本地名称合在一起有一个全局惟一的名字，就是我们知道的***限定名（qualified name）***。

命名空间声明一般出现在元素的起始标记中，通常习惯于把一个命名空间的名称映射为另一个通常很短的字符串，也就是我们所知的***命名空间前缀（namespace prefix）***。命名空间声明的语法为：xmlns: prefix='URI'。无前缀使用默认命名空间声明映射命名空间也是可能的。默认命名空间声明的语法是：xmlns='URI'。在这两种情况下，URI 都出现在单引号（'）或者双引号（"）中。一个元素只能有一个默认命名空间声明出现。但是非默认的命名空间声明则是不限数目的，并且可以为相同的 URI 提供不同的前缀部分。尽管把相同 URI 映射到多个前缀并不是特别有用，但它是合法的。

所有的命名空间声明都有作用域，这个作用域就是它适用的一套元素。命名空间声明的作用域是它声明的元素以及元素的后代。在作用域内给后代元素所使用的前缀提供一个新的映射可以取代给定前缀到命名空间名称的映射。给后代元素提供默认的命名空间声明可以取代在作用域内的默认命名空间。

文档里的所有符合 XML 规范中命名空间的元素名称叫作 QName。从语法上来说，所有 QName 都有本地名和可选前缀。本地名和前缀合称为 NCName。NCName 是一个不带冒号的名字。有前缀的元素的语法是：前缀，后面跟着冒号，然后跟着本地名。赋予前缀的元素的命名空间就是作用域内为该前缀声明所指定的命名空间。如果在作用域里没有这样的命名空间声明就是错误的。如果无前缀元素的命名空间存在的话，该命名空间是作用域

内默认的命名空间声明所指定的命名空间。如果作用域中没有默认命名空间声明，则该元素不存在于任何命名空间中。不存在于任何命名空间中的元素被称作***非限定元素***（***unqualified elements***）。非限定元素的命名空间名称是空字符串“”。如果默认命名空间声明在作用域中，并且需要有非限定元素，则可以为元素提供 xmlns= ‘’ 这样的形式来屏蔽默认命名空间声明。

例子

限定的元素名和非限定的元素名

```
<pre:Person xmlns:pre='urn:example-org:People' >
  <name>Martin</name>
  <age>33</age>
</pre:Person>
```

例子中元素的本地名叫作 Person，前缀 pre 是从命名空间 urn：example-org：People 映射来的。元素 Person 有两个后代，本地名为 name 和 age，它们都是非限定元素，因为它们不存在于任何的命名空间中。

使用默认命名空间声明的限定和非限定元素

```
<Person xmlns='urn:example-org:People' >
  <name xmlns=''>Martin</name>
  <age xmlns=''>33</age>
</Person>
```

元素的本地名叫作 Person，没有前缀。根据作用域内该 URI 的默认命名空间声明元素存在于命名空间 urn：example-org：People 里。元素有两个后代，本地名分别叫作 name 和 age。这两个子元素都是非限定的，也就是说，他们不存在于任何的命名空间里。这个例子和前面的例子是等同的。

限定元素

```
<pre:Person xmlns:pre='urn:example-org:People' >
  <pre:name>Martin</pre:name>
  <pre:age>33</pre:age>
</pre:Person>
```

该元素的本地名是 Person，前缀是 pre，是从命名空间 URI urn：example-org：People 映射来的。元素有两个本地名为 name 和 age 的后代。这两个子元素都有同一个前缀 pre，都存在于命名空间 urn：example-org：People。

使用默认命名空间声明的限定元素

```
<Person xmlns='urn:example-org:People' >
  <name>Martin</name>
  <age>33</age>
</Person>
```

元素的本地名叫作 Person，没有前缀。根据作用域内该 URI 的默认命名空间声明元素存在于命名空间 urn：example-org：People 里。元素有两个本地名为 name 和 age 的后代。这两个子元素都存在于命名空间 urn：example-org：People 之中。这个例子和前面的例子是等同的。

1.3 属　　性

```
name='value'
name="value"
```

元素可以用属性来注释。属性通常用来编码实际数据或者为元素提供元数据，也就是说，提供关于元素所显示内容的额外信息。给定元素的属性被串行化在元素的起始标记中。属性表示为用等号（=）分开的名/值对。属性名与元素名有相同的构造规则。属性值是自然的文本形式，必须出现在单引号或者双引号的内部。元素可以有任意数目的属性，但是它们的名称必须不同。

例子

数据属性

```
<Person name='Martin' age='33' />
```

本例中的 Person 使用属性来表示要比使用子元素更加清楚。

元数据属性

```
<age base='16' units='years' >20</age>
<age base="10" units="years" >32</age>
```

带有元数据属性的元素。

1.4 属性和命名空间

```
prefix:localname='value'
prefix:localname="value"
```

属性名称都是 QName。赋予前缀的属性的命名空间就是作用域内为该前缀声明所指定的命名空间。如果在作用域里没有这种命名空间声明是错误的。即使在作用域中有默认命名空间声明，无前缀属性也是不存在于任何命名空间中的。

例子

限定属性

```
<Person xmlns='urn:example-org:People'
        xmlns:b='urn:example-org:People:base'
        xmlns:u='urn:example-org:units' >
  <name>Martin</name>
  <age b:base='10' u:units='years' >33</age>
</Person>
```

本地名叫作 base 的属性存在于命名空间 urn：example-org：People：base 中；本地名叫作 units 的属性存在于命名空间 urn：example-org：units 中。