



XML与WML 整合应用

隆华软件工作室 编著

本书内容包括：

- XML 的语法介绍
- WML 语法介绍
- 如何建立 WAP 网站
- 提供一些编辑 WAP 网页的工具软件



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



XML 与 WML 整合应用

隆华软件工作室 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书介绍了 XML 及其应用, 包括 XML 的标记语法、DTD 文件格式、CSS 排版功能、XLS 及 DOM; 然后介绍了无线应用协议 (WAP) 及其应用, 包括 Card 元素、变量与 Task 元素、文字格式与字符引用、控制元素以及网站链接元素等。每一章都包含有大量实用的范例代码。读者阅读本书后, 马上就能设计自己的网页, 建立自己的 WAP 网站。

本书适用于网站与网页设计人员、网站编辑、网站管理人员以及移动网络用户阅读。

本书繁体字版名为《精通 XML 与 WML 整合应用》, 由文魁资讯股份有限公司出版, 版权属隆华软件工作室所有。本书简体字中文版由文魁资讯股份有限公司授权清华大学出版社独家出版。未经本书原版出版者和本书出版者书面许可, 任何单位和个人均不得以任何形式或任何手段复制或传播本书的部分或全部。

北京市版权局著作权合同登记号: 图字 01-2000-4399 号

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

书 名: XML 与 WML 整合应用

作 者: 隆华软件工作室

责任编辑: 郭福生

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>

印 刷 者: 北京市昌平环球印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787 × 1092 1/16 印张: 14.25 字数: 236 千字

版 次: 2001 年 5 月第 1 版 2001 年 5 月第 1 次印刷

书 号: ISBN 7-302-04406-6/TP · 2591

印 数: 0001 ~ 5000

定 价: 19.90 元

序

近年来，随着因特网逐渐成为主要的通信网络，我们的日常生产活动的很大一部分就是处理描述顾客群、需求、购买模式及如何建立顾客所需产品的信息。因此随着 Web 的爆炸性成长，使得单纯用来展示网页内容的 HTML 已经无法满足我们的需要。因此，一种通用的标准格式就应运而产生了，那就是 XML。

XML 格式应用在因特网上，而另一种现今最流行的格式就是 WAP。WAP 和 XML 不同的地方是 WAP 应用于移动电话。本书详细说明了 XML 及 WAP 的相关语法，希望能给读者以相当的帮助。

如果您已经非常熟悉 XML 格式，可以直接跳到第 7 章学习 WML；如果您对 XML 不太熟悉，请按顺序阅读各章。

隆华软件工作室

目 录

第1章 XML 简介

1.1 什么是 XML	1
1.2 XML 文件的结构与格式	1
1.3 常用的 XML 编辑工具	2
1.4 结论 2	2

第2章 XML 的标记语法

2.1 XML 的声明	3
2.1.1 编码声明	3
2.1.2 元素声明	4
2.1.3 声明元素的排列	4
2.1.4 根节点与子节点	5
2.1.5 属性声明	6
2.1.6 指定所要处理的语言	7
2.1.7 命名规则	8
2.1.8 注释	8
2.1.9 字符引用	9
2.1.10 预先定义的实体	9
2.2 XML 文件的实体结构	10
2.2.1 XML 文件的逻辑结构	10
2.2.2 文件的实体结构	11
2.2.3 可分解实体	11

2.2.4	不可分解实体	12
2.2.5	使用实体应注意的事项	12
2.3	CDATA 区	13
2.4	使用中文内容或中文标记名称	14

第3章 DTD- 文件

3.1	DTD 文件格式	17
3.1.1	内部 DOCTYPE 声明	17
3.1.2	外部 DOCTYPE 声明	18
3.2	引用私有的 DTD 文件	19
3.3	引用公有的 DTD 文件	20
3.4	声明元素	21
3.5	定义元素	22
3.5.1	以 ANY 关键词来定义元素	23
3.5.2	以 EPTY 关键词来声明元素的数据类型	24
3.5.3	以 #PCDATA 关键词来定义元素	26
3.5.4	指定元素声明及定义的顺序	29
3.5.5	以控制标记来定义元素出现的次数	31
3.5.6	以选择性的方式来定义元素	32
3.5.7	以混合式的方式来定义元素	33
3.6	声明属性	34
3.7	属性类型	36
3.7.1	CDATA 类型	36
3.7.2	列举类型	36
3.7.3	NMTOKEN 类型	38
3.7.4	NMTOKES 类型	39
3.7.5	ID 类型	41

3.7.6 IDREF 类型	42
3.7.7 IDRES 类型	44
3.8 设置属性的方式	45
3.8.1 #REQUIRED	46
3.8.2 #IMPLIED	46
3.8.3 #FIXED	47
3.8.4 直接设置默认值	49
3.9 实体	50
3.9.1 内部实体的声明	50
3.9.2 外部实体的声明	52
3.9.3 参数型的实体	54
3.10 调试功能	56
3.10.1 IGNORE	56
3.10.2 NCLUDE	57

第4章 CSS 排版功能

4.1 如何在XML中使用CSS文件	59
4.1.1 在XML中引用CSS文件	59
4.1.2 在XML中将DTD文件和CSS文件合并使用	60
4.2 文字边框及设置	62
4.3 段落设置	67
4.4 背景设置	70
4.5 区块功能	73
4.6 定位功能	76

第5章 XLS

5.1 如何在XML中引用XSL文件	79
5.2 如何直接获得XML元素值	79

5.3	如何取得多个 XML 元素值	81
5.4	多重样式版	84
5.5	排序	86
5.5.1	升序	86
5.5.2	降序	89
5.6	查找特定的值	91
5.6.1	等于条件式	92
5.6.2	不等于条件式	94
5.6.3	大于条件式	97
5.6.4	小于条件式	100
5.7	查询	102
5.7.1	如何以属性值来作为查询的条件	102
5.7.2	xsi:if 查询	105
5.7.3	xsl:choose:si:when 和 xsi:otherwise	107

第 6 章 DOM

6.1	Node 对象	111
6.2	NodeList 对象列表	118
6.3	document 对象	122
6.4	Element 对象	128
6.5	Attr 对象	131

第 7 章 WAP 简介

7.1	WAP 简介	134
7.2	常用的 WAP 工具	136
7.3	NOKIA WAP TOOLKIT 软件的安装	137
7.4	如何在 Microsoft IIS 中架设 WAP 网站	141

第 8 章 卡片组和卡片

8.1	Card 元素	149
8.2	head 元素及 access 元素	150
8.3	template 元素	151
8.3.1	do 元素	151
8.3.2	onenterforward 元素与 onevent 元素	153
8.3.3	onenterbackward 元素与 onevent 元素	155
8.3.4	ontimer 元素与 onevent 元素	157
8.4	onpick 元素	159

第 9 章 变数及 Task 元素

9.1	变量的命名规则	161
9.2	设量变量	161
9.3	Task 元素	168
9.3.1	go 元素	168
9.3.2	prev 元素	170
9.3.3	refresh 元素	172
9.3.4	noop 元素	173

第 10 章 文字用格式及字符引用

10.1	br 元素	176
10.2	p 元素	177
10.3	table 元素	179
10.3.1	tr 元素	183
10.3.2	td 元素	183
10.4	em 元素	183
10.5	b 元素	184
10.6	i 元素	185

10.7	u 元素	186
10.8	字符引用	187

第 11 章 控制元素

11.1	select 元素及 option 元素	189
11.2	select 元素的 iname 及 ivalue 属性	191
11.3	多重选择	194
11.4	optgroup 元素	196
11.5	input 元素	198
11.6	输入密码	202
11.7	fieldset 元素	206

第 12 章 网站链接元素及 timer 元素

12.1	anchor 元素	208
12.2	a 元素	212
12.3	timer 元素	215

第 1 章 XML 简介

1.1 什么是 XML

XML (eXtensible Markup Language 可扩展标记语言) 是一套用来设定原则并允许用户自定义类似 HTML 的标记语言, 以方便数据的存取、处理、交换与转换等操作。以往的 HTML, 是用来制作供展示用的网页的。随着 Web 的不断发展, 使得单纯用来展示网页内容的 HTML 已经不能满足用户的需要。举例来说, 当我们要在网页上要求用户输入基本资料或者是筛选资料时, HTML 就无能为力了。

XML 以简单且标准的方式将各种原始信息, 如文字、电子表格和数字, 以标记的方式进行储存, 因此用户可以根据自己的需要来定义。XML 的另一个优点是支持中文, 也正因为如此, 在最近几年中, XML 几乎成为每一位网络 IT 人士必学的利器。在未来几年内, 将会逐渐看到 XML 被大量应用在各式各样的数据交换环境中, 尤其是在电子商务等相关领域。在电子商务领域中, 由于各家的电子商务系统并无固定统一的规范, 而造成上游厂商与中、下游厂商的整合出现断层, 而这种断层的救星非 XML 莫属。

1.2 XML 文件的结构与格式

XML 文件必须符合 Valid 和 Well-formed 这两个格式中的一个。所谓的 Well-formed 是一种比较简单的格式, 因为它只需确定在 XML 文件中所有的标记都有起始标记和结束标记就可以了。如果要使用一个空的元素, 则要确保在空的元素中放一个 (/) 符号, 并且由 XML 处理器来进行验证。而另一种格式是 Valid 格式, 这种格式的 XML 除了必须符合 Well-formed 所规定的条件之外, 还必须包含 DTD (文档类型定义) 文件。

在处理 XML 文件时, 需要有一个 XML 处理器, 而这个 XML 处理器的功能与解释器一样, 除了要能够阅读 XML 文件之外, 还要能够把数据传给应用程序。关于如何读取和应该传递哪些数据的各项参数, 已经包括在 XML 规范当中。

1.3 常用的 XML 编辑工具

1. XML SPY: 也是本书所使用的编辑工具。XML SPY 的操作方法非常简单, 初学者不需要借助于操作手册即可使用。XML SPY 软件是基于 Windows 95/98 的, 在 Windows NT 上运行也没有问题。
2. SAXON: 此套软件是用来处理 XML 文件的, 其主要支持的组件包括: XSLT 处理器、XSLT 1.0 及 XPath 推荐标准、.Java 函数库等。可以从站点 <http://users.iclway.co.uk/mhkay/saxon/instant.html> 下载此套软件。
3. Xalan-Java: 此套软件最大的功能就是可以将 XSLT 格式快速转换, 只需要一条指令就可以把 XML 文件转换成需要的文件格式, 如 HTML 或纯文本。它也支持由 W3C 认证的 XSLT 1.0 和 XPath 1.0 推荐标准, 可以说它是一个非常健全的参考工具。可以从 <http://xml.apache.org/xalan/getstarted.html#download> 下载此套软件。

1.4 结论

XML 的出现将互联网的发展带入一个系统整合的新纪元, 它使不同的平台或系统能够相互通信。而 XML 的使用范围从文字处理、电子商务到数据备份, 几乎无处不在。XML 的影响力可说是令人生畏, 各大厂商无不争相推出与 XML 技术相关的产品, 以求在未来的系统整合市场上, 占有一席之地。关于这一点, 可以从 Microsoft 最近发布的 SQL Server 2000 以及 BizTalk Server 上看出端倪。可以相信, 在不久的将来, XML 将会扮演一个十分重要的角色。

第2章 XML 的标记语法

XML 文件是由声明、元素、指令以及注释组成的。在开始学习 XML 的应用之前，首先要学习在 XML 中如何声明标题、元素、属性以及命名规则等相关主题。毕竟，要学习如何跑，首先应该学会如何走路！

2.1 XML 的声明

在每一个 XML 文件开头，都要进行声明，格式如下：

```
<?xml version="1.0"?>
```

这个声明定义了 XML 的版本，并且在编写时必须使用小写字母。可以从范例 2.1 来了解此声明在 XML 文件声明的位置。

范例 2.1

```
<?xml version="1.0"?> {XML 文件的版本声明}  
  
<note>  
  
  <to>Tove</to>  
  
  <from>Jani</from>  
  
  <heading>Reminder</heading>  
  
  <body>Don't forget me this weekend!</body>  
  
</note>
```

2.1.1 编码声明

在 XML 声明中，除了可以声明版本之外，还可以声明编码。编码声明是用来标识编码的，不同的编码方式使用不同的字符或语言。在 XML 中，默认的编码语言是 UTF-8 或 UTF-16。在 XML 中要进行编码声明的方式如下：

```
<?xml version="1.0" encoding='UTF-8'?>
```

或

```
<?xml version="1.0" encoding='UTF-16'?>
```

如果要使用中文解码,应该如何下达指令呢?其实,如果要以中文的方式来解码,可以按照下述方式来下达指令:

```
<!xml version="1.0" encoding="GB2312"?>
```

但是,并不是每一个XML文件都支持GB2312码,但每一个XML文件都支持UTF-16码和UTF-8码。

2.1.2 元素声明

在XML中,元素的声明由3部分组成:起始标记、元素内容和终止标记。3部分缺一不可。其中,起始标记由<、元素名称和>组成;元素内容则是由一连串的字符组成;而终止标记由</、与起始标记相同的元素名称以及>组成。一个标准的XML元素声明如下所示:

```
<mailname>test@test.com.tw</mailname>
```

其中,<mailname>是起始标记,test@test.com.tw是元素内容,</mailname>是终止标记。注意,起始标记的名称与终止标记的名称一定要相同,尤其是名称的大小写所代表的意义是不同的。下面的范例是错误的:

```
<Mailtest>test@test.com.tw</mailtest>
```

如上例所示,其起始标记(<Mailtest>)与终止标记(</mailtest>)的名称不完全一样,因此会产生错误。

2.1.3 声明元素的排列

一般来说,在一个XML文件中,不会仅有一个元素声明,通常都是同时声明许多元素。因此,当有多个元素要声明时,其排列方式一定要按照元素的顺序依次来声明,也就是说,元素间的声明标记不能相互重叠。下面的范例就是一个正确的排列方式:

```
<STNO>

  <NO>1107</NO>

  <NAME>John</NAME>

  <SEX>Male</SEX>

</STNO>
```



1. 在程序的第1行中, 声明一个根节点(所谓根节点, 将在后面的章节中说明) <STNO> 的起始标记。而 </STMO> 终止标记则放在程序的最后一行。
2. 在程序的第2, 3, 4行中, 声明了 NO、NAME 和 SEX 等元素。这些元素的声明都符合必须有起始标记及终止标记的规定, 而且各元素之间的起始标记和终止标记并没有相互重叠。

范例 2.2 是一个错误的范例。

范例 2.2

```
<STNO>

    <NO>1107

    <NAME>John

    <SEX>Male</NAME></SEX>

</STNO>
```



1. 在程序第2行中, NO 元素的声明并没有包含终止标记, 因此会产生错误信息。
2. 在程序第4行中, NAME 元素声明的终止标记与 SEX 元素声明的起始标记相互重叠。这样的排列方式不符合声明元素的起始标记和终止标记不能相互重叠的规定, 因此会产生错误信息。

2.1.4 根节点与子节点

在 XML 文件中, 只有一个根节点, 换句话说, 也就是在 XML 文件中仅能存在一个根元素。所谓的根元素就像文件目录结构中的根目录一样, 也可以说和 HTML 文件中的 <HTML> 标记一样, 根元素只能有一个。当然, 一个根元素内可以包含多个子元素, 而在子元素内又可以包含其他的子元素, 这种结构与文件目录结构是相似的。在文件目录结构中, 其根目录内可以包含多个子目录, 而这些子目录又可以包含其他的子目录。现在, 就下面的范例来说明:

```
<STNO>

  <NO>1107</NO>

  <NAME>JOHN LAIO</NAME>

    <FIRST>JOHN</FIRST>

    <LAST>LAIO</LAST>

</STNO>
```

在上述范例中，STNO 是根节点，相当于文件目录结构中的根目录，位于最上层；而 NO 和 NAME 是 STNO 的子元素，也就是子节点。NAME 子元素内又包含 FIRST 和 LAST 两个元素，这时 NAME 元素又变成 FIRST 和 LAST 这两个元素的父元素，也就是父节点，而 FIRST 和 LAST 两个节点是 NAME 元素的子节点。

2.1.5 属性声明

当声明了元素之后，接着就可以指定元素的属性了。属性的声明方式如下：

```
<!element.name property="value">
```

或

```
<!element.name property='value'>
```

上述式子的说明如表 2.1 所示。

表 2.1 符号与说明

符号	说明
<	起始标记
element.name	元素名称
property	属性名称
=	等于
"	双引号
'	单引号
value	属性值
>	终止标记

范例 2.3

```
<custom name="John">

<custom tel='02-1234567'>
```


在范例 2.3 中, custom 是元素的名称, 而 name 和 tel 是属性的名称, "John" 和 '02-1234567' 是 custom 元素的属性值。

在声明属性的时候, 可以使用双引号或单引号来指定属性值, 但千万不要混合使用, 如下例所示的声明是不允许的:

```
<custom name="John">
```

2.1.6 指定所要处理的语言

当要指定 XML 所要处理的语言时, 可以使用属性: xml:lang。其声明方式如下:

```
<xml:lang="language code - country code">
```

其中, language code 是 ISO 639 所规定的语言代码, 并且要用小写字母来表示; 而 country code 是指 ISO 3166 中所规定的国家代码, 并且要用大写字母来表示。表 2.2 及表 2.3 分别列出了 language code 及 country code 所代表的意义。

表 2.2 ISO 639 语言代码

代码	语言
ar	阿拉伯语
ch	中文
de	德语
en	英语
es	西班牙语
fr	法语
gr	希腊语
it	意大利语
ja	日语
nl	荷兰语
pt	葡萄牙语
ru	俄语

表 2.3 ISO 3166 国家代码

代码	国家	代码	国家
AT	奥地利	BE	比利时
CA	加拿大	CN	中国
DE	德国	DK	丹麦
EN	英格兰	ES	西班牙
FR	法国	GR	希腊
IT	意大利	JA	日本
NL	荷兰	PG	葡萄牙
RU	俄罗斯	US	美国