

## 附录D IE5 XML模式和数据类型

虽然使用文档类型定义（Document Type Definition，DTD）能够成功地定义XML文档，但是我们似乎希望通过更加灵活的方式定义XML文档的结构。另外，在XML文档的设计阶段，最好能够指定数据类型，以便尽量将XML文档的处理工作机械化。

为此，W3C正在——在编写本书时——从事有关XML模式和数据类型的建议的制定。其中包括提议的文档内容定义（Document Content Definition，DCD）语言。正如本附录所介绍的，Internet Explorer 5从相当程度上支持XML模式和数据类型的标准实现。IE5正在开发这项技术，当它正式发布时，附录中列出的属性有可能被修改。

### D.1 XML模式

XML模式（XML Schema）是对XML文档结构的描述或定义。模式本身是用XML编写的。DTD要求使用者学习类似SGML的语法，与此相比，模式的这一特性使得它更易于被初学者理解。

Internet Explorer 5实现的XML模式提供了8个预定义的元素，它们可以用来定义XML文档：

表 D-1

名 称	描 述
Schema	模式的整体包含元素，它定义了模式的名称
ElementType	定义模式中使用的元素类型
element	定义已声明的元素的实例，它用在<ElementType>元素中
AttributeType	定义模式中使用的属性类型
attribute	定义已声明的属性的实例，它用在<ElementType>元素中
datatype	定义属性或元素可以包含的数据类型
description	用于提供有关属性或元素的信息
group	用于组合若干元素，以定义特定的元素序列

#### D.1.1 IE5 XML模式的元素

本节按照字母顺序详细描述了XML模式的所有元素，以及它们的属性。

##### 1. attribute元素

<attribute>元素用于定义特定的属性实例，它可以用在<AttributeType>或<ElementType>元素中。

表 D-2

元素名称	属 性	描 述
attribute	default	属性的缺省值，当required的值为“no”时，可以指定default的值。如果required为“yes”，则文档中提供的属性值必须与缺省值相同

(续)

元素名称	属 性	描 述
	required	指定属性值是否是必需的。它的值可以是“ yes ”或“ no ”
	type	指定<AttributeType>，本属性是它的实例

2. AttributeType元素

<AttributeType>元素用于定义模式中的元素使用的属性的类型。可以使用 <attribute>元素进一步指定属性的特定实例。

表 D-3

元素名称	属 性	描 述
AttributeType	default	属性的缺省值。如果属性是枚举类型的，则该缺省值必须从列表中 选择
	dt:type	属性可以接受的数据类型
	dt:values	构成枚举类型的一组值，例如：“ roses carnations daisies ”
	name	在模式中标识<AttributeType>元素的唯一字符串，它同时定义了 属性的名称
	model	定义属性是否能够接受模式中未定义的内容。值“ open ”表示 允许出现未定义的内容，值“ closed ”表示只允许出现模式中定 义的内容
	required	指定属性值是否是必需的。它的值可以是“ yes ”或“ no ”。当 required为“ yes ”时，它与default是互相排斥的

<AttributeType>元素中的dt:type和dt:values与<datatype>元素中对应的属性具有相同的使用方法：

```
<AttributeType name="flowername"
  default="rose"
  dt:type="enumeration"
  dt:values="rose carnation daisy lilac" />
```

需要注意的是，虽然dt是数据类型常用的命名空间前缀，但是我们也可以使用其他前缀。

3. datatype元素

<datatype>元素用于定义属性或元素可以包含的数据类型。在编写本书时，对本元素的支持相当有限。

表 D-4

元素名称	属 性	描 述
datatype	dt:max	元素或属性可接受的最大值（包含边界值）
	dt:maxExclusive	元素或属性可接受的最大值（不包含边界值），即：元素或 属性的值必须小于指定的值
	dt:maxlength	元素或属性的值的最大长度。对于字符串，它指定了字符 数。对于数字和二进制值，它指定了存储数据所需的字节数
	dt:min	元素或属性可接受的最小值（包含边界值）。

(续)

元素名称	属 性	描 述
	dt:minExclusive	元素或属性可接受的最小值 ( 不包含边界值 ), 即: 元素或属性的值必须大于指定的值
	dt:type	本附录最后列出的特殊数据类型或简单数据类型。
	dt:values	对于 enumeration 类型, 它包含枚举值的列表。

#### 4. description 元素

<description>元素用于提供有关属性或元素的信息。

表 D-5

元素名称	属 性	描 述
description	无	元素或属性的描述性文本

#### 5. element 元素

<element>元素用于定义 <ElementType> 元素中使用的元素的特定实例。

表 D-6

元素名称	属 性	描 述
element	type	本模式或其他模式中定义的元素类型的名称, 本元素是该元素类型的实例
	minOccurs	通过模式定义元素在文档中是否是可选的。“0”表示元素是可选的, 可以不出现; “1”表示元素至少出现一次。缺省值为“1”
	maxOccurs	通过模式定义元素在文档中允许出现的最大次数。“1”表示只能出现一次; “*”表示可以出现多次。缺省值为“1”

#### 6. ElementType 元素

<ElementType>用于定义模式中使用的元素的类型。可以通过 <element>元素进一步指定元素的特定实例。

表 D-7

元素名称	属 性	描 述
ElementType	content	定义了元素能够包含的内容的类型。“empty”意味着无内容; “textOnly”意味着它只能包含文本 ( 除非 model 为 “open” ); “eltOnly”意味着只能包含其他元素, 不能包含文本; “mixed”意味着它能够包含任何内容
	dt:type	本附录最后列出的特殊数据类型或简单数据类型
	model	定义元素是否能够接受模式中未定义的内容。值 “open” 表示允许出现未定义的内容, 值 “closed” 表示只允许出现模式中定义的内容
	name	在模式中标识 <ElementType> 元素的唯一字符串, 它同时定义了元素的名称

(续)

元素名称	属 性	描 述
	order	定义了元素出现的顺序。值“one”意味着在包含的一组element元素中，只能出现其中的一个元素；值“seq”意味着包含的所有元素必须按指定顺序出现；值“many”意味着包含的元素可以选择是否出现以及出现的次数和顺序

下面的7“group元素”中提供了有关content和order属性的例子。

7. group元素

<group>元素用于将若干<element>和/或<attribute>元素组合在一起，并给它们设置特定的顺序。利用模式中的<group>元素，能够精确地控制元素和/或属性在文档中出现的顺序。

表 D-8

元素名称	属 性	描 述
group	minOccurs	通过模式定义组在文档中是否是可选的。“0”表示组是可选的，可以不出现；“1”表示组至少出现一次。缺省值为“1”
	maxOccurs	通过模式定义组在文档中允许出现的最大次数。“1”表示只能出现一次；“*”表示可以出现多次。缺省值为“1”
	order	定义了本组包含的其他组和元素类型出现的顺序。值“one”意味着在所包含的组和元素类型中，只能出现其中一个；值“seq”意味着包含的所有组和元素类型必须按指定顺序出现；值“many”意味着包含的组和元素类型可以选择是否出现以及出现的次数和顺序

下面的例子说明了如何利用组和元素类型定义元素在文档中出现的次数和顺序。

程序清单 D-1

<pre>&lt;ElementType name="first" content="empty" /&gt; &lt;ElementType name="second" content="textOnly" dt:type="string" /&gt; &lt;ElementType name="thirdEqual" content="empty" /&gt;  &lt;ElementType name="third" content="eltOnly" order="many"&gt;   &lt;element type="thirdEqual" /&gt; &lt;/ElementType&gt;  &lt;ElementType name="fallen" content="empty" /&gt; &lt;ElementType name="unplaced" content="empty" /&gt; &lt;ElementType name="last" content="empty" /&gt;  &lt;ElementType name="raceorder" order="seq"&gt;    &lt;element type="first" /&gt;   &lt;element type="second" /&gt;   &lt;element type="third" /&gt;    &lt;group minOccurs="1" maxOccurs="1" order="one"&gt;     &lt;element type="fallen" /&gt;     &lt;element type="unplaced" /&gt;   &lt;/group&gt; &lt;/ElementType&gt;</pre>	
--	--

```
<element type="last" />
</group>

</ElementType>
```

由于主元素 raceorder 定义了属性 order="seq"，因此，<first>、<second>和<third>元素必须按照指定的顺序至少出现一次。group 元素也受到同样的限制；然而，对于组中定义的三个元素，只有一个能够出现在文档中。所以，下面的组合都是合法和有效的。

程序清单 D-2

```
<first />
<second>too slow again</second>
<third />
<fallen />
<first />
<second />
<third>
  <thirdEqual />
</third>
<unplaced />
<first />
<second>still too slow</second>
<third>
  <thirdEqual />
  <thirdEqual />
  <thirdEqual />
</third>
<last />
```

8. Schema 元素

<Schema>元素是用于包含模式的元素。它定义了模式的名称，以及模式使用的命名空间。

表 D-9

元素名称	属 性	描 述
Schema	name	定义了模式的名称，模式是通过该名称被引用的
	xmlns	指定了模式中元素和属性的缺省命名空间的 URI
	xmlns:dt	指定了模式中 datatype 属性的命名空间的 URI

```
<Schema name="myschema"
  xmlns="urn:schemas-microsoft-com:xml-data"
  xmlns:dt="urn:schemas-microsoft-com:datatype">
```

正如我们前面所说的，datatype 命名空间前缀不一定是 dt，但 dt 是常用值，而且它能够明确指出（对于人类来说）使用该前缀的属性属于 datatype 命名空间。然而，此处使用的前缀必须与命名空间定义（的 URN 部分）相同。

D.1.2 IE5 XML 模式的结构

以下代码显示了 IE5 XML 模式的完整结构，其中包含每个属性值应有的类型。虽然元素可以出现在多个位置，但是在后面出现的元素中，我们删除了属性列表，以免过多的重复：

## 程序清单 D-3

```

<Schema name="schema_name"
  xmlns="namespace_URI"
  xmlns:dt="namespace_URI" >

  <AttributeType default="default_value"
    dt:type="xml_data_type"
    dt:values="enumerated_value_list"
    name="name_or_id"
    model="open"|"closed"
    required="yes"|"no">

    <datatype dt:max="maximum_value"
      dt:maxExclusive="maximum_value_exclusive"
      dt:maxLength="maximum_length"
      dt:min="minimum_value"
      dt:minExclusive="minimum_value_exclusive"
      dt:type="xml_data_type" />
      dt:values="enumerated_value_list" />

    <description>description_text</description>

  </AttributeType>

  <AttributeType>
    ... etc ...
  </AttributeType>

  <ElementType content="empty"|"textOnly"|"eltOnly"|"mixed"
    dt:type="xml_data_type"
    model="open"|"closed"
    name="name_or_id"
    order="one"|"seq"|"many" >

    <description>description_text</description>

    <datatype ... etc ... />

    <element type="element_type"
      minOccurs="0"|"1"
      maxOccurs="1"|"*" />

    <attribute default="default_value"
      required="yes"|"no" />

    <attribute ... etc ... />

    <group minOccurs="0"|"1"
      maxOccurs="1"|"*"
      order="one"|"seq"|"many" >

      <attribute ... etc ... />

      <element ... etc ... />

    </group>

  </ElementType>

</Schema>

```

## D.2 XML数据类型

数据类型是通过数据类型命名空间引用的，该命名空间是在模式的 XML <Schema>元素中使用xmlns:datatype命名属性声明的。

下表显示了 W3C 提出且 Internet Explorer 5 支持的数据类型，它包含所有常用的数据类型，以及 SQL、Visual Basic、C、C++ 和 Java 等流行的数据库、编程语言和系统内置的类型。本表选自 <http://www.w3.org/TR/1998/NOTE-XML-data/> 上的 W3C 注释。

表 D-10

名 称	解析类型	存储类型	举 例
string	pcdata	字符串 (统一码)	Ομωνύμα λεγάται αν ονομα α μονον κοι νον, ο δε κατ α του νομα λογος της ουσ ιας ετερος, οι ον ζυον ο τε ανθρωπος και το γε γραμμενον.
number	对各位数没有限制的数字，可以有前导符、小数和指数 标点符号采用 US English 的格式	字符串	15, 3.14, -123.456E+10
int	数字，可以有符号，但不能包含小数或指数	32位有符号 二进制数	1, 58502, -13
float	与“number”相同	64位IEEE 488	.314159265358979E+1
fixed.14.4	与“number”相同，但是在小数点左侧至多有14位，右侧至多有4位	64位有符号 二进制数	12.0044
boolean	“1”或“0”	位	0, 1 (1=="true")
dateTime.iso8601	ISO 8601格式的日期，可以包含时间，但不能包含时区、秒的小数部分可以精确到纳秒	包含年、月、日、时、分、秒和纳秒的结构或对象	19941105T08:15:00301
dateTime.iso8601.tz	ISO 8601格式的日期，可以包含时间和时区。秒的小数部分可以精确到纳秒	包含年、月、日、时、分、秒、纳秒和时区的结构或对象	19941105T08:15:5+03
date.iso8601	ISO 8601格式的日期。(不能包含时间)	包含年、月和日的结构或对象	19541022
time.iso8601	ISO 8601格式的时间，不能包含日期和时区	包含时和分的结构或对象	

(续)

名 称	解析类型	存储类型	举 例
time. iso8601.tz	ISO 8601格式的时 间，不能包含日期， 但是可以含时区。	包含时、分、 时区时和时区分 的结构或对象	08:15-05:00
i1	数字，可以有符号， 不能含小数或指数	8位二进制数	1, 255
i2	同上	16位二进制数	1, 703, -32768
i4	同上	32位二进制数	
i8	同上	64位二进制数	
ui1	无符号数，不能包 含小数或指数。	8位无符号 二进制数	1, 255
ui2	同上	16位无符号 二进制数	1, 703, -32768
ui4	同上	32位无符号 二进制数	
ui8	同上	64位无符号 二进制数	
r4	与“number”相同	IEEE 488 4字节 浮点数	
r8	同上	IEEE 488 8字节 浮点数	
float. IEEE.754.32	同上	IEEE 754 4字节 浮点数	
float. IEEE.754.64	同上	IEEE 754 8字节 浮点数	
uuid	代表字节的十六进制 数，可以有内置的连 字符，且连字符应该 被忽略	128字节Unix UUID结构	F04DA480-65B9-11d1- A29F-00AA00C14882
uri	统一资源标识符	遵循W3C规范	<a href="http://www.ics.uci.edu/pub/ietf/uri/draft-fielding-uri-syntax-00.txt">http://www.ics.uci.edu/pub/ietf/uri/draft-fielding-uri-syntax-00.txt</a> <a href="http://www.ics.uci.edu/pub/ietf/uri/">http://www.ics.uci.edu/pub/ietf/uri/</a> <a href="http://www.ietf.org/html.charters/urn-charter.html">http://www.ietf.org/html.charters/urn-charter.html</a>
bin.hex	代表字节的十六进 制数	没有指定长度	
char	字符串	一个统一码字符 (16位)	
string.ansi	只包含<= 0xFF的 ASCII字符的字符串	统一码或单字节 字符串	

在上表中，所有iso8601xxx形式的日期和时间实际上都使用了ISO 8601定义的格式的有限子



集。如果指定年，它必须是四位的。日期格式中不采用序数日期。对于使用星期数的格式，必须要截去年和月。

## XML简单数据类型

W3C还推荐了XML 1.0中使用的记号化数据类型。它们有时被称为简单类型。Internet Explorer 5支持以下简单类型：

表 D-11

名 称	描 述
entity	XML ENTITY类型
entities	XML ENTITIES类型
enumeration	枚举类型，例如：允许值列表
id	XML ID类型
idref	XML IDREF类型
idrefs	XML IDREFS类型
nmtoken	XML NMTOKEN类型
nmtokens	XML NMTOKENS类型
notation	NOTATION类型
string	代表普通的String数据类型