

# XML 在 ASP 服务端的 应用研究

杨厚群 (海口市海南大学信息学院 570228)

摘要	本文通过采用一个在线教程的实例,通过在ASP服务端操纵XML,讨论了利用DOM以及DTD技术对XML文档结构的处理,可以控制向客户端发送数据,使服务端对数据中间件的操纵更加合理和便捷。具体从DOM对树型结构的处理加以分析。
关键词	XML DOM DTD

## 1 引言

随着新版本的浏览器(如IE5.0、NN5.0)的发布,XML的使用在Web应用领域更加普及和更加便捷,如何利用XML最新的DOM(文档对象模型)在服务端分析和使用ASP应用程序中的XML数据是处理客户与服务端、服务端与服务端之间数据交换的核心环节。

由于DOM具备了在服务端分析和使用XML的能力,为开发者提供了全新的功能。随着XML使用范围的急剧扩大,在服务端操纵XML的应用需求也随着增加。

## 2 DOM对XML文档的处理

最新的DOM在IE5中完全支持W3C关于DOM(Level 1)建议中所描述的编程接口,它包括了多种支持XML相关技术(诸如XSL、XSL套匹配、命名空间、数据类型和规划)的新方法,DOM本质上是一个XML解析器,DOM把XML文档当作易于导航和使用的树形结构来进行处理,其在应用程序中的处理过程如图1所示。

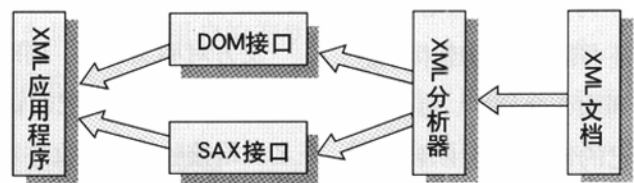


图1 应用程序开发过程中的DOM

DOM为开发者提供的访问XML文档信息的媒介是一种分层对象模型,而这个层次的结构,则是根据XML文档生成的节点树。一个XML分析器,在对XML文档进行分析之后,不管这个文档有多简单或者多复杂,其中的信息都会被转化成对象节点树。在这个节点树中,有一个根节点(即Document节点),所有其他的节点都是根节点的子节点。节点树生成之后,就可以通过DOM接口访问、修改、添加、删除、创建树中的节点和内容。

在文档对象树中,文档中所有的内容都是用节点来表示的。一个节点又可以包含其他节点,节点本身还可能包含一些信息,例如节点的名字、节点值、节点类型等。文档中的根实际上也是一个元素,之所以要把它单独列出来,是因为在XML文档中,所

有其他元素都是根元素的子元素,而且根元素是唯一的,具有其他元素所不具有的某些特征。

文档对象模型利用对象来把文档模型化,这些模型不仅描述了文档的结构,还定义了模型中对象的行为。换句话说,节点不是数据结构,而是对象,对象中包含方法和属性。在DOM中,对象模型要实现:

- (1) 用来表示、操作文档的接口;
- (2) 接口的行为和属性;
- (3) 接口之间的关系以及互操作。

在DOM接口规范中,有四个基本的接口: Document, Node, NodeList以及NamedNodeMap。在这四个基本接口中,Document接口是对文档进行操作的入口,它是从Node接口继承过来的。Node接口是其他大多数接口的父类,象Document, Element, Attribute, Text, Comment等接口都是从Node接

口继承过来的。NodeList 接口是一个节点的集合，它包含了某个节点中的所有子节点。NamedNodeMap 接口也是一个节点的集合，通过该接口，可以建立节点名和节点之间的一一映射关系，从而利用节点名可以直接访问特定的节点。

为了在 ASP 应用程序中的服务端操纵 XML，本文将使用一个跨平台站点上有关网络教学的在线教程的 XML 版本，剖析如何创建一个完整的 ASP 页面模型，显示在线课程名称和发布日期以及对相关课程的讨论，并显示所有在线课程和讨论的链接。

在服务端的 ASP 应用程序中使用 DOM 并不困难，但前提是需要服务端安装 IE5，因为大量的支持组件会随着 IE5 安装到机器上，一旦 IE 安装到服务端，需要做的事就是创建 DOM 对象，下面是创建 DOM 对象的代码：

```
<% Set objXML = Server.CreateObject("Microsoft.XMLDOM") %>
```

只要在服务端创建了 DOM 对象，就可以创建自己的 XML 文档或装入已存在的 XML 文档，在装入文档时，服务端就已经具备了装入 XML 文本流或打开 XML 文档取出内容的功能。在整个应用模型中。假定在服务端中已有在线教程的 XML 文档的副本。在装入文档前，将 DOM 对象的 syncs 属性值设为：“False”，这会告诉 DOM 对象不允许异步下载 XML 文档。这点非常重要的，因为文档装入后立即处于使用状态，如果这时内容没装入或没完全装入，操作者很可能在接下来的操作中得到错误的提示。

```
<% objXML.async = False
objXML.Load (Server.MapPath("onlinecourse.xml"))
%>
```

下面是实际操作的 XML 文档

```
<?xml version="1.0"?>
<!DOCTYPE onlinecourse
SYSTEM
"http://www.classic.net.cn/dtd/
courseware.dtd">
<course>
<header>
<copyright>Copyright 2000-2001
通信与信息系统
</copyright>
<coursewareVersion>2.0</
coursewareVersion>
<pubDate>2000 年 6 月 1 日</
pubDate>
<lastBuildDate>2001 年 6 月 30
日</lastBuildDate>
</header>
<item>
<text>XML 在线教程</text>
<link>
<url>http://www.classic.net.cn/
course/xml/001.htm</url>
<linetext>XML </linetext>
</link>
</item>
?</pre>
```

```
<item>
<text>Java 在线教程</text>
<link>
<url> http://www.classic.net.cn/
course/Java/001.htm</url>
<linetext>Java</linetext>
</link>
</item>
</course>
```

### 3 出错解析与数据发布

#### 3.1 出错解析

DOM 对象为开发者提供了一个出错解析对象，其间包含了最新的出错解析信息。这个对象对调试和纠错特别有用。本文建议在操作过程中，一旦装入了 XML 文档，便立即检查所有的出错解析对象。

```
<% If objXML.parseError.
errorCode <> 0 Then
handle the error
End If
%>
```

非常幸运的是，解析对象提供给开发者以下有关错误的许多有用的信息

错误码	描述
filepos	包含错误的 XML 文档中，文件的绝对位置
Line	XML 文档中发生错误的行号
Linepos	错误行中包含错误的具体字符位置
reason	造成错误的原因
srctxt	发生错误的数据位置
url	包含错误的 XML 文档的统一资源定位符

在本文在线课程模型中。由于 XML 文档由 DTD 进行校验，出错解析对象就有了更为深刻的意义。在这种情况下，不仅 XML 文档需要格式良好，而且为了避免出错，必须要与 DTD 相对应，在装入 XML 文档后注意检查出错解析对象是一种有益的操作。

前面的合法的例子都在文档的序言包含了 DTD。但是 XML 真正的功能来自于不同的人编写的可为许多文档共享通用的 DTD。如果 DTD 不是直接包含在文档内，而是由外部联结而来，则 DTD 的改变会自动传播给使用它的所有文档。另一方面，当 DTD 改变时并不能确保其向后兼容性。不兼容的改

变会破坏文档。

尽管大多数文档由易于定义的部分组成,但不是所有的文档都使用共同的模板。许多文档为自己使用而增加特定元素时,可能需要象 HTML4.0 DTD 这样的标准 DTD。其他文档可能只使用标准元素,但需要对它们重新排序。例如,一个 HTML 主页可能有一个 BODY 元素,它必须包含一个 H1 标题标记后接一份 DL 定义列表,而另一个 HTML 主页可能有一个 BODY 元素,它包含许多不同的顺序不定的标题标记、段落和图象。如果特定的文档与同一站点上其他页面具有不同的结构,在文档本身内定义结构比在单独的 DTD 中定义更实用。这种方法也使文档更易于编辑。

为了实现这样的目的,文档可使用内部和外部 DTD。内部声明放在 <!DOCTYPE> 标记尾部的方括号中。例如,假设需要一个包括在线课程统计并有页眉和页脚的主页。这样的文档可如下面代码所示。课程信息可从文档 COURSE.DTD 中得到,构成外部 DTD 子集。基本元素 DOCUMENT 以及元素 TITLE 和 SIGNATURE 的定义来自包含文档中的内部 DTD 子集。通常,更为通用的部分可能应该是外部 DTD 的一部分,而内部内容则更是与特定专题有关。

现在 DOM 对象中已经有了格式良好、有效的文档,再看看在文档中有写什么,DOM 提供了大量有用的方法,能精确地判定 XML 文档中包含的所有东西,因为 DOM 显示了作为节点树(由一个元素和所有下级元素组成)的文档大纲,为了操纵数据我们将结束创建一系列节点对象。本文模型将使用 getElementBytagname 方法得到文档中的元素列表或节点。

### 3.2 数据发布

当前首要任务是找到在线教程的

发布日期。通过检查 DTD,可以知道这个信息存储在发布日期节点上,一种简单处理节点内容的办法是先创建一个针对 XML 文档中所有节点的节点显示对象,然后遍历所有节点直到找到发布日期节点,由于 DTD 指定了发布日期节点不包含任何子节点,所以我们可以使用文本属性,立即取出节点内容。

```
<%
    Set objLst = objXML.
    getElementByTagName("*")
    For i = 0 to (objLst.length ? 1)
        If objLst.item(i).nodeName =
        "pubDate" Then
            StrDate = objLst.item(i).text
        Exit For
    End If
Next
%>
```

模型在上述代码中通过 getElementBytagname 方法,由它返回了一个包含 XML 文档所有节点(或元素)的节点列表对象,由于使用 DTD 可以得到发布日期节点的精确位置,所以可以直接使用它的编号来寻址,正如本文在上例中已做的那样。由于节点列表是组织在一起的,

遍历整个文档是一种非常有效的手段。

现在已经有了发布日期,如何找出文档里访问量较大(即点击数、回复数较大)的在线教程,可以重复一次,通过 DTD 取回在线课程所处的项目节点,在文档中每一门课程都有一个项目节点,再遍历一次,就象上面做的那样,每过一节点,计数器加 1。有一种更好的办法去找回这条信息,就是使用 DOM 提供的。

在上面的模型中,开发者所做的事情是创建一个包含所有项目节点的节点列表对象,我们可以使用 length

属性来查找出在节点列表对象到底有多少节点,换句话说,有多少门课程在线。

```
<% Set objLst = objXML.
    getElementByTagName("item")
    strNoOfHeadlines = objLst.
    Length
    %>
    正如平时在 ASP 页面来显示一些这样信息一样,下列的代码告诉开发者如何通过遍历头条节点列表在 ASP 页面中显示所有课程的 URL
    <HTML>
    <BODY>
    <H1>Courseware Headlines</H1>
    <% For i = 0 To (noOfHeadlines ?
    1)
        Set objHdl = objLst.item(i)
        Response.Write("<a href="" & _
            objHdl.childNodes(1).
            childNodes(0).text & _
            """">" & objHdl.
            childNodes(0).text & _
            "</a><br>")
    Next
    %>
    </BODY>
</HTML>
```

## 4 结论

深刻了解 XML 文档结构信息和利用 DOM 的功能,开发者可以十分便捷地解析在 ASP 服务端的 XML 文档,然后给客户端发送相关数据,上述的实例能够在几乎所有浏览器上使用,这对于日趋活跃的 XML 充当数据中间件这种角色的普遍应用来说,实现信息的搜集和发布,为开发者所拥有的解决方案的优化提供了一个全新的途径。■