返回总目录

# 目　　录

# 第五部分　基于 XML 的 Web 站点应用与开发

# 第十八章　WIDL 自动控制 Web 站点

本章介绍新一代的 Web 接口定义语言 WIDL，通过对其基本概念、特性、优点、规范、元素说明的描述以及一个具体实例的讲解，使读者了解并学会应用这种崭新的网络自动化控制技术。

本章包括以下内容：

- WIDL 概述
- 规范与元素说明
- 应用实例

## 18.1　概　　述

Web 接口定义语言 WIDL（Web Interface Definition Language）是一个基于 XML 的接口定义语言（IDL），它允许自动完成同 Web 文件和表格的交互操作，提供根据标准的 Web 协议完成请求/应答交互操作的通用方法，或允许将 Web 作为一种通用的集成平台使用；它的目的是促进 Intranet 或者 Internet 上的商业应用之间的数据交换。WIDL 提供了 Web 自动控制的基础，Web 自动控制是一种允许 Web 浏览器以外的其他应用直接和 Web 服务器、存储在 Web 服务器上的数据进行互操作的技术。通过 WIDL，交易就可以在一个广泛的商业应用和公司的 Web 服务器上建立直接的连接。这个方法利用了 Web 的标准协议，允许大量的应用访问和共享存储在某个地方的相同资源。

在今天，诸如“一个电子表格、常用数据库这样的商业应用通过 Internet 或者 Intranet 彼此交换数据”此类在非 Web 应用里的数据交换和交流的工作，已经由成千上万行的程序代码、大量的 Web 站点管理员，以及富有创造性的程序开发人员实现。其过程是把一个应用的数据转换成 Web 信息，然后把这些数据通过 Intranet 或 Internet 传输，最后把它转换成接收方的数据类型。对于共享商业应用数据来说，这显然不是最简单和最好的方法。

现在大多数的数据库软件，都提供将存储在它们里面的数据连接到 Web 页面的工具。除了在 Web 页面中包含来自于数据库的数据之外，Web 页面还经常被用来自动更新数据库，如：在 Web 页面上使用表格收集信息。WIDL 把这个功能扩展到了所有的商业应用，而且允许用一种标准的方法，与存储在 Web 服务器上的数据相交流。

为 WIDL DTD 而开发的文档，描述了一系列的服务或功能，比如跟踪一个数据包，或者定义一个人员配置文件（Personal Profile）。任何能理解和处理 WIDL 的软件包，都能够执行一个服务，然后为用户显示结果。这样通过 WIDL，就不再需要 Web 浏览器浏览 Web 页面了。任何兼容 WIDL 的软件包都能够处理和显示 Web 数据。

WIDL 并不是为某个 Web 开发者而设计的方案，相反，它是特别为或大或小的商业需求而设计的，可以和其他的组织共享数据。WIDL 提供了一种存储和显示数据的新方法，而且它还利用了现存的软件，允许大量的应用去共享同一个数据。比如在现在的市场上，有几种相互

竞争的家庭财务软件包，包括 Microsoft 的 Money 和 Intuit 的 Quicken，这两个软件包描述的是相同的信息（帐目记录或类似的东西，但是使用不同的、私有的数据格式。缺少从另一个应用类型输入和转换数据的机制，Money 不能以自己的格式读取 Quicken 的文件，同样 Quicken 也不能读取 Money 的文件。

现在我们假设有一家银行想通过 Web 给它的客户提供帐目记录的服务，如果要使这些帐目记录既能够被 Money 读取，也能够被 Quicken 读取，那么该银行除了需要提供 Web 版本的帐目记录之外，还不得不为这两种应用程序提供两种格式的文件。这种状况就是当前银行所面临的实际问题，如果银行采用了支持 Money 和 Quicken 的 WIDL，而且对词汇表做了某些调整，那么问题就会得到解决，因为这两个软件包都能够从一个单一资源里读取数据。此外，因为 WIDL 是用来允许任何应用都能够读取和显示 Web 数据的，因此客户可以使用 Money 或在 Quicken 来直接查看他们的在线账目数据，而不必使用 Web 浏览器。

WIDL 的潜在应用前景是无限的。现在世界上已经有了成千上万的独立数据管理系统，大多数都不能彼此对话。如果某个公司购买的是某一种数据管理系统，而他们的人力资源和客户数据却保存在其他两个不同的系统中，会出现什么情况？他们就不得不将这些系统里的一种数据格式转换成另外一种数据格式。对于商业应用，这并不是非常有效的办法。

如果读者认为 WIDL 可以用来很好地解决不同领域、组织之间如何以一种标准的方式共享数据的问题，并对它非常感兴趣的话，那么可以访问 WebMethods 中的 WIDL 主页 http://www.Webmethods.com。WebMethods 是 WIDL 和 Web 自动控制技术的创造者，在该站点里包含了各种各样的演示和白皮书，论述了 B2B 的 WIDL 实现。Web 自动控制工具包是 WebMethods 的 WIDL 开发工具，这是一个实现 Web 自动化的优秀应用。

WebMethods 公司开发 WIDL 的最初目的，是为了给万维站点提供 API，以便应用系统可以通过编程来访问万维网。因此，WIDL1.x 和 2.x 规范都定义了这种语言，该语言既指定了接口，又定义了接口规范映射到万维站点上的方法。WIDL3.0 将接口规范和文档映射实现方案，分别放到各自的 XML 文档中。因此 WIDL3.0 定义了两个组件：一个 IDL 组件和一个文档映射组件。同时使用这两个组件，应用系统就可以在网络上进行通信，而不用考虑应用系统是用什么编程语言编写的，也不用考虑 XML 应用系统符合哪个 DTD。

我们先来看看 WIDL3.0 的 IDL 组件。下例显示了一个完整的 WIDL3.0 接口规范的例子。

```
<WIDL NAME="buying_wealth" VERSION="3.0">
<RECORD NAME="Subscribe">

  <VALUE NAME="OrderID" TYPE="A3"/>
  <VALUE NAME="Property"/>
  </RECORD>
<RECORD NAME="SaleAccept">
<VALUE NAME="OrderNumber" TYPE="A3"/>
<VALUE NAME="Wealth"/>
<VALUE NAME="AccountBalance" TYPE="T1"/>
</RECORD>
<METHOD NAME="OrderWealth" INPUT="OrderWealth"
        OUTPUT="OrderAccept" RETURN="OrderNumber"/>
</WIDL>
```

即使 WIDL 对于某些人来说有些超前，但它表明了 XML 将如何被用来既为工业界，也为个人用户使用。实际中存在需求，希望各种不同的软件包能够通过 Web 以一种更好的、更有效率的方法来共享数据，而 WIDL 提供了满足这个需求的解决方案的基础。今天，大多数发展中的 XML 接口定义语言，都是为了迎合某种特定的需求而设计的。用户可以为自己的需求找到最好的方案。

## 18.2 规范与元素说明

WIDL 已经在 1997 年 9 月 22 日提交给 W3C，作为一个草案，WIDL 还不是一个标准，但是它确实有很大希望成为标准。注意 W3C 的 XML 页面 http://www.w3.org/XML/，关注一下任何有关 WIDL 的最新消息。

注　在 W3C 的站点 http://www.w3.org/TR/NOTE/widl 里，可找到全部 WIDL 草案的内容。

1．规范

WIDL DTD 既短小又精简，它只有 6 个元素。WIDL 是用来协调应用的，WIDL 的大多数功能都固定在这些应用里面。WIDL 只不过描述了一个服务，而这些应用知道该如何使用它们。

WIDL DTD

```
<!ELEMENT WIDL ( SERVICE | BINDING )* >
<!ATTLIST WIDL
      NAME          CDATA     #IMPLIED
      VERSION (1.0 | 2.0 | ...) "2.0"
      TEMPLATE     CDATA     #IMPLIED
      BASEURL      CDATA     #IMPLIED
      OBJMODEL (wmdom | ...) "wmdom"
>

<!ELEMENT SERVICE EMPTY>
<!ATTLIST SERVICE
      NAME          CDATA     #REQUIRED
      URL           CDATA     #REQUIRED
      METHOD (Get | Post) "Get"
      INPUT         CDATA     #IMPLIED
      OUTPUT        CDATA     #IMPLIED
      AUTHUSER      CDATA     #IMPLIED
      AUTHPASS      CDATA     #IMPLIED
      TIMEOUT       CDATA     #IMPLIED
      RETRIES       CDATA     #IMPLIED
>

<!ELEMENT BINDING ( VARIABLE | CONDITION | REGION )* >
<!ATTLIST BINDING
```

```
        NAME            CDATA     #REQUIRED
        TYPE (Input | Output) "Output"
>

<!ELEMENT VARIABLE EMPTY>
<!ATTLIST VARIABLE
        NAME            CDATA     #REQUIRED
        FORMNAME     CDATA     #IMPLIED
        TYPE (String | String[] | String[][]) "String"
        USAGE (Default | Header | Internal) "Function"
        REFERENCE    CDATA     #IMPLIED
        VALUE          CDATA     #IMPLIED
        MASK           CDATA     #IMPLIED
        NULLOK                   #BOOLEAN
>

<!ELEMENT CONDITION EMPTY>
<!ATTLIST CONDITION
        TYPE    (Success | Failure | Retry) "Success"
        REF             CDATA     #REQUIRED
        MATCH          CDATA     #REQUIRED
        REBIND        CDATA     #IMPLIED
        SERVICE       CDATA     #IMPLIED
        REASONREF    CDATA     #IMPLIED
        REASONTEXT CDATA     #IMPLIED
        WAIT           CDATA     #IMPLIED
        RETRIES       CDATA     #IMPLIED
>

<!ELEMENT REGION EMPTY>
<!ATTLIST REGION
        NAME           CDATA     #REQUIRED
        START          CDATA     #REQUIRED
        END            CDATA     #REQUIRED
>
```

2. 元素

WIDL 的 6 个元素能够用来描述任何类型的一个或者一系列的服务，从销售记录列表到商品类型描述都可以。WIDL 是一种基于内容的标记语言，是用来描述供计算机阅读和处理的内容的。下面我们举例说明每个元素该如何用来描述信息。

（1）WIDL 元素。WIDL 元素就是任何 WIDL 文档元素，元素的内容和属性规范包括：

- 内容：SERVICE 和 BINDING 元素的一个或多个实例。
- 属性：NAME,UERSION,TEMPOATE,BASEURL,LBJMOKEL。

这两个元素 SERVICE 和 BINDIMG，以及它们的 CHILD 元素，都必须嵌套在 WIDL 元素里面。WIDL 元素带有的属性见表 18-1。

在这些属性当中，最重要的就是 NAME 和 VERSION 属性，下面是应用它们的例子：

```
<WIDL  NAME=〝Book Sale〞  VERSION=〝1.0〞>
</WIDL>
```

表 18-1

| 属性 | 说明 | 类型 | # 缺省 |
|---|---|---|---|
| NAME | 为接口声名一个名称 | String | Exactly One |
| VERSION | 指定用来描述接口的 WIDL 版本 | String | 0 or 1 "2.0" |
| TEMPLATE | 指明服务必须遵守的特定规范或一套指令 | URI | 0 or 1 |
| BASEURL | 指明用来描述该接口及其服务的主要 URL | URI | 0 or 1 |
| OBJMODEL | 指明对象模型，该对象模型将支配文档元素被应用识别和显示的方法 | String | 0 or 1 wmdom |

（2）SERVICE 元素。SERVICE 元素描述接口的一个特定服务（一个请求或者一个应答）。该元素的内容和属性包括：

- 内容：空
- 属性：NAME、URL、METOD、INPUT、OUTPUT、AUTHUSER、AUTHPASS、TIMEOUT、RETRIES

多数的接口可能会拥有多个服务，然而也有只包含一个服务的。服务的实际数目，完全由接口的目的来决定。

SERVICE 元素的属性包括：用来指明服务的名字的 NAME 属性；用来指向 Web 上该服务及其信息存放在什么位置的 URL 属性；用来指明该服务是否是通过 POST 或 GET 方法来处理的 METHOD 属性；用来指定服务和 Web 服务器上的数据交互（发送和接收数据）所必需的绑定，或者方法和变量的 INPUT、OTPUT 属性，接口和数据类型的需求将决定每个服务的绑定，或者方法和变量的 AUTHUSER 和 AUTHPASS 属性，接口和数据类型需求将决定每个服务的绑定；用来存储用户名口令信息和属性，几乎所有的客户端都会提示用户名和口令；用来指明在客户端放弃和服务器之间的连接之前，有多长时间客户端必须连接的 TIMEOUT 属性；此外还有用来指明客户端可以有多少次试图连接服务器的 RETRIES 属性。

在这些属性当中，最重要的就是 URL、INPUT、OUTPUT。书籍销售记录服务的应用实例如下：

```
<WIDL  NAME=〝Book Sale〞  VERSION=〝1.0〞>
 <SERVICE  NAME=〝account〞  METHOD=〝post〞
          URL=http://www.bhp.com/accouts/data
          INPUT=〝getaccount〞
          OUTPUT=〝saleaccount〞
          TIMEOUT=〝60〞
          RETRIES=〝6〞
</WIDL>
```

（3）BINDING 元素。BINDING 元素包含用来提供有关绑定和变量的其他元素，这些绑定和变量是一个服务完成它的工作所必需〞的。该元素的内容和属性包括：

- 内容：UARIABLD、CINDITION 或 REGION 元素的一个或多个实例。

- 属性：NAME、TYPE。

属性为绑定信息分配一个名字，而 TYPE 属性则指明该绑定是一个输入，还是一个输出的绑定。帐目服务是非常简单的，Web 服务器可以基于简单的服务描述，给应用提供所有的信息。

（4）VARIABLE 元素。VARIABLE 元素嵌套在 BINDING 元素里面，当服务用来和 Web 服务器通信时，它用来为服务提供信息的详细的内容。该元素的内容和属性包括：

- 内容：空
- 属性：NAME、FORMNAME、TYPE、USAGE、REFERNCE、VALUE、NULLOK 和 MASK

VARIABLE 元素拥有下面的关键属性：

- NAME：指明变量的名称。
- FORMNAME：给服务器提供通过 GET 或 POST 的方法提交的值，GET 和 POST 方法在 SERVICE 元素里已经指明了。
- TYPE：指明变量的数据类型和维数。
- USAGE：指定该变量是否应该作为标题信息传递给服务器，是否应该用在 WIDL 应用的里面。
- VALUE：指定变量的值。

BINDING 元素可以根据应用或服务的需要，包含许多 VARIABLE 元素。因为书籍销售记录服务没有 BINDING 标志，所有它也不需要 VARIABLE 元素。

（5）CONDITION 元素。元素定义了当一个绑定成功者失败了时，会发生什么。该元素的内容和属性包括：

- 内容：空
- 属性：TYPE、REF、MATCH、REBIND、SERVICE、REASONREF、REASONTEXT、WAIT、RETRIES

当一个服务事件成功或者失败了时，这些属性提供了会发生什么事情的细节。TYPE 属性指明了在服务的成功或失败或重试的事件中，REBIND 属性可以用来将服务重新引导到另外版本的绑定中，而 REASONTEXT 属性则表示当服务失败了时，将会返回的一个文本串。同样的情况，因为书籍销售记录服务中没有 BINDING 元素，所以它也没有 CONDITION 元素。

（6）REGION 元素。在 WIDL DTD 中的最后一个元素是 REGION 元素。这个元素是用来将一个 WIDL 绑定的输出嵌入其他的 XML 或 HTML 文档中的。这意味着 WIDL 并不只是单独应用，而是可以和其他依赖于 HTML 和 XML 的工具和文档一起发挥作用的。该元素的内容和属性包括：

- 内容：空
- 属性：NAME、START 和 END

NAME 属性为该 REGION 元素提供一个容易辨认的名字，而 START 和 END 属性则表明了在什么地方，这些绑定输出必须包含在文档里面。区域信息会随变量和条件信息的不同，根据基于绑定的基础被指定。书籍销售记录服务是用来独立工作的，因此它没有包含区域信息。

## 18.3 应 用 实 例

### 18.3.1 WIDL Examples

下面的例子说明了如何使用 WIDL 为通用的导航定义一个跟踪记录服务。通过允许 WIDL 定义引用一个模板的 WIDL 定义，我们可以定义许多浏览导航服务。FoobarShipping 是实现 Shipping 接口的方法之一。

```
<WIDL NAME="genericShipping" TEMPLATE="Shipping"
        BASEURL="http://www.shipping.com" VERSION="2.0">

<SERVICE NAME="TrackPackage" METHOD="Get"
            URL="/cgi-bin/track_package"
            INPUT="TrackInput" OUTPUT="TrackOutput" />

<BINDING NAME="TrackInput" TYPE="INPUT">
    <VARIABLE NAME="TrackingNum" TYPE="String" FORMNAME="trk_num" />
    <VARIABLE NAME="DestCountry" TYPE="String" FORMNAME="dest_cntry" />
    <VARIABLE NAME="ShipDate" TYPE="String" FORMNAME="ship_date" />
</BINDING>

<BINDING NAME="TrackOutput" TYPE="OUTPUT">
    <CONDITION TYPE="Failure" REFERENCE="doc.title[0].text"
                MATCH="Warning Form" REASONREF="doc.p[0].text" />
    <CONDITION TYPE="Success" REFERENCE="doc.title[0].text"
                MATCH="Foobar Airbill:*" REASONREF="doc.p[1].value" />
    <VARIABLE NAME="disposition" TYPE="String" REFERENCE="doc.h[3].value" />
    <VARIABLE NAME="deliveredOn" TYPE="String" REFERENCE="doc.h[5].value" />
    <VARIABLE NAME="deliveredTo" TYPE="String" REFERENCE="doc.h[7].value" />
</BINDING>

</WIDL>
```

在上例中，绑定在 TrackInput 的值通过 HTTP 的 Get 方法作为命名值指向定位在 http://www.shipping.com/cgi-bin/track_package 的服务。TrackOutput 绑定中使用了对象引用来检测服务是否成功完成，文档中的元素值可通过 HTTP 请求返回。

WIDL 是用来扩展 Web 服务器的功能的，它允许 Web 服务器上的数据用大量多种多样的应用去浏览。下一章我们将探讨如何将 Web 的内容传送到用户端的一些方法：CDF，它的全称是频道定义格式，用来将 Web 的内容基于一个有规则的时间表，通过 Web 频道传送给用户。

## 18.4 小　　结

WIDL 为符合不同 DTD 的 XML 文档之间提供了一个接口规范，应用 WIDL 可以对应用接口和 XML 消息 DTD 做透明的映射，简化应用系统，充分发挥 XML 的优势，使应用系统具有最高程度的易访问性，更好地解决系统的互操作性问题。

# 第十九章　频道定义格式推送 Web 站点

本章关注 CDF 技术，它为 Web 站点发布带来光明而美好的前景。这种允许网络开发人员向用户自动发送新的网站内容与信息的机制，直接导致了最早的 XML 词汇表的出现，它可以说是 XML 应用中最为成熟的产品之一。通过 CDF，我们可以比较深入地了解 XML 的具体应用情况以及技术向产品转化的过程。

本章包括以下内容：

- 认识 CDF
- CDF 文档规范及频道创建
- CDF 高级应用

## 19.1　认识 CDF

频道定义格式（Channel Definition Format，CDF）是一种用于定义频道并基于 XML 的标记语言。频道允许 Web 站点自动通知阅读者修改重要信息。类似于订购服务，这种方法也称为 Webcasting 或者推送。

CDF 文件是一个与站点中的 HTML 文档分开但是链接到 HTML 文档的 XML 文档。在 CDF 文档中定义频道建立阅读者和站点内容之间的连接参数。数据可以通过推送（发送通知或整个 Web 站点到注册的阅读者）或者拉回（阅读者选择地在他们的 Web 浏览器中加载 Web 页并得到修改信息）来传送。

频道的类型有 3 种：

- 阅读者在他们的桌面上看的拉回频道。
- 通过电子邮件通知阅读者修改信息的推送频道。
- 发送已修改页面到阅读者以便离线浏览的推送频道。

人们能够通过预订站点，将这个站点下载到他们的家庭或者公司的计算机。通过下载此站点，用户可以在闲暇的时间进行预览，避免在观看站点时等待链接到 Internet 的时间。用户不必每次来访问固定的站点就可以自动获得由网站自动发送的最新资源，它还提供了拨号用户离线浏览的功能。只要用户预先订阅所需的频道，并设置好更新时间，就可以获得由浏览器自动下载、更新的资料，WIN98 中可以设置的活动桌面，就是 IE 的一种频道浏览方式。如果用户自己的站点使用这种技术，就可以使站点轻松具有专业水准的风格，并可以大大方便站点的访问者。

首先让我们来看一个具体的例子：

First.cdf:

```
< ?XML VERSION="1.0" ENCODING="gb2312"? >

< CHANNEL HREF="http://www.mycdf.com/main.htm" >
    < ABSTRACT >CDF 学习站< /ABSTRACT >
```

< /CHANNEL >

我们可以发现，频道文件和 HTML 文件一样，也是用一种标签来实现各种设置的。上述文件的第一行表示，符合 XML（扩展标注语言）标准，版本是 1.0，支持的字符集是 gb2312，它的第二行用 CHANNELHREF 标签定义了一个频道，并且指出了它的地址。在括号< ABSTRACT >与< /CHANNEL >之间定义了用户将鼠标移到频道栏上所出现的提示文字。最后的< /CHANNEL >标签标志着频道文件的结束。

CDF 文件的标签有 24 个，每一个都有复杂的设置，以下是一些典型的符号，我们可以很容易地将它们用在自己的频道文件之中。

A〈A HREF=〝URL〞〉　〈/A〉：定义一个超级链接；

ABSTRACT < ABSTRACT >〝提示信息〞< /ABSTRACT >：显示提示信息；

CHANNEL 〈CHANNEL HREF=〝URL〞〉...〈/CHANNEL〉；定义一个频道；

ITEM　〈ITEM HREF=〝URL〞...〈/ITEM〉：定义一个子频道；

LOGO　〈LOGO HREF=〝URL〞STYLE=〝ICON〞|〞 I〝MAGE〝〉：　显示图标或者图像；

TITLE　< TITLE >标题< /TITLE >：频道标题；

下面例子非常简单，因为没有网络连接，所以运行时仅仅只是打开一个 IE 窗口并且显示 c:\windows 目录的内容，也可以把频道与自己的站点连接起来，只需要修改 URL 即可。

```
< ?XML VERSION="1.0" ENCODING="gb2312"? >
< CHANNEL HREF="file:/ c:/windows /" >
< TITLE >studio< /TITLE >
< LOGO HREF="file:/ c:/windows /Circles.bmp" STYTLE="IMAGE"/ >
< ABSTRACT >"Connect me!"< /ABSTRACT >
< /CHANNEL >
```

## 19.2　CDF 文档规范及频道创建

频道是一个定期更新、定期通知的 Web 站点，它是采用了一种被称为推送的技术。频道最主要的内容是 CDF，该文件是一个文本文件，遵循扩展标记语言 XML 的规范。

### 19.2.1　CDF 文档规范说明

1．频道元素的子元素

每一个 CHANNEL 元素可以包括各种子元素。表 19-1 列出了可能的子元素和其含义。

表 19-1　通道的可能子元素

| 标记 | 含义 |
|---|---|
| LASTMOD | 上次修改页面的时间 |
| TITLE | 通道名称 |
| ABSTRACT | 通道的简洁描述 |
| AUTHOR | 作者 |
| PUBLISHER | 出版者 |
| LOGO | 可以显示给用户表示通道的图像 |

（续表）

| 标记 | 含义 |
| --- | --- |
| A | 被推送的页面，不提倡 HREF 属性 |
| SCHEDULE | 当通道应该刷新时，告诉浏览器的信息 |
| LOGTARGET | 登录文件应该上载到的 URL |
| LOGIN | 表示在用户修改内容之前，浏览器应该询问用户的用户名称和口令 |

相关的 DTD 如下：

```
<!ELEMENT LastMod EMPTY>
<!ATTLIST LastMod VALUE CDATA #REQUIRED>
<!ELEMENT Title EMPTY>
<!ATTLIST Title VALUE CDATA #REQUIRED>
<!ELEMENT Abstract EMPTY>
<!ATTLIST Abstract VALUE CDATA #REQUIRED>
<!ELEMENT Author EMPTY>
<!ATTLIST Author VALUE CDATA #REQUIRED>
<!ELEMENT Publisher EMPTY>
<!ATTLIST Publisher VALUE CDATA #REQUIRED>
<!ELEMENT Copyright EMPTY>
<!ATTLIST Copyright VALUE CDATA #REQUIRED>
<!ELEMENT PublicationDate EMPTY>
<!ATTLIST PublicationDate VALUE CDATA #REQUIRED>
<!ELEMENT Keywords EMPTY>
<!ATTLIST Keywords VALUE CDATA #REQUIRED>
<!ELEMENT Category EMPTY>
<!ATTLIST Category VALUE CDATA #REQUIRED>
<!ELEMENT Rating EMPTY>
<!ATTLIST Rating PICS-Label CDATA #REQUIRED>
```

（1）内容描述。定义的每一个通道都有一个标题和一个摘要。通道的标题与页面的标题不一样，通道标题出现在通道指南、通道列表和通道栏中。

如果包括了通道信息，那么阅读者就可以查看这种信息。描述包含在通道元素的子元素<ABSTRACT>中。该摘要元素还包含 PCDATA。每一个通道和项目可以有自己的标题和描述。描述出现在工具提示窗口中，具体实现程序代码如下。因为工具提示有限的空间，所以摘要最多可以有 200 个字符。然而在 DTD 中，不要求长度限制。

```
<?xml version="1.0"?>
<CHANNEL HREF="http://www.writelivelihood.com/newscover.html">
    <TITLE>UserAble</TITLE>
    <ABSTRACT>The Monthly 'zine from Write Livelihood</ABSTRACT>
    <CHANNEL HREF="http://www.writelivelihood.com/column.html">
        <TITLE>Currents and Trends</TITLE>
        <ABSTRACT>A look at current concerns, trends, and issues in the technical communication field.
    </ABSTRACT>
    </CHANNEL>
    <CHANNEL HREF="http://www.writelivelihood.com/tips.html">
```

```
        <TITLE>Tip of the Month Club</TITLE>
        <ABSTRACT>
    Clearing stumbling blocks from the field of
    endeavour. Contributions from various sources.
  </ABSTRACT>
    </CHANNEL>
</CHANNEL>
```

（2）LOGOS。LOGO 元素把图像与通道联系起来。元素是<CHANNEL>元素的子元素。对每个通道和项目元素最多可以增加 3 个 Logo。

在通道内，Logo 可以用在许多不同的方式：桌面上的图标、程序发射器中的图标以及 MS 通道指向和通道栏中的 Logo。Internet Explorer 和 Logo 支持 GIF、JPEG 和 ICO 格式图像但是不是活泼的 GIF。因为用图标可以以桌面上的整个颜色范围和图案出现，所以使用 GIF 作为图标的透明背景。通过与 Web 站点的外观和感觉相匹配的通道 Logo，可以在阅读者的桌面上、通道栏上和通道指南上维持商标外观。

LOGO 元素有两个属性，HREF 和 STYLE。样式属性允许有 ICON、IMAGE 和 IMAGE-WIDE Logo。这些 logo 是不同大小和类型的图像，如表 19-2 所示。

表 19-2　LOGO 元素的 STYLE 属性值

| 图像类型 | 描　　述 |
|---|---|
| ICON | 显示层次中子元素旁边的文件列表和通道栏中的 16×16 像素图标。 |
| IMAGE | 显示在桌面通道栏中的 80（宽）×32（高）像素图像。 |
| TMAGE-WIDE | 显示在浏览器中的 194（宽）×32（高）像素图像。如果通道层次嵌套在下面，那么当阅读者单击该 logo 时，它们出现。 |

下例中嵌套的 CDF 文档使用各种大小的 Logo。

```
<?xml version="1.0"?>
<CHANNEL HREF="index.html" BASE="http://www.writelivelihood.com/">
        <TITLE>Write Livelihood</TITLE>
        <ABSTRACT>Write Livelihood provides technical documentation solutions
    for companies large and small.</ABSTRACT>
        <LOGO HREF="logo_icon.gif" STYLE="ICON"/>
        <LOGO HREF="corp_logo_regular.gif" STYLE="IMAGE"/>
        <LOGO HREF="corp_logo_wide.gif" STYLE="IMAGE-WIDE"/>
        <CHANNEL>
                <TITLE>UserAble</TITLE>
                <ABSTRACT>Write Livelihood's Monthly magazine with product
        reviews, industry news, and personal views.</ABSTRACT>
                <LOGO HREF="logo_icon.gif" STYLE="ICON"/>
                <ITEM HREF="editorial.html">
                        <TITLE>Currents and Trends</TITLE>
                        <ABSTRACT>The view from our perch or is that porch?
          </ABSTRACT>
                        <LOGO HREF="logo_icon.gif" STYLE="ICON"/>
                </ITEM>
```

```
    </CHANNEL>
</CHANNEL>
```

当建立通道时，可以只为最上层的通道或者为任何通道或者在 CDF 中定义的项目定义 IMAGE 和 IMAGE-WIDE logo。不能保证 IMAGE-WIDE logo 用作该层次的次要元素。

当为浏览器通道栏设计 logo 时，记住对内容的修改可以在 logo 图像的左上角进行加亮。这样就隐藏了左上角的内容。

另外，如果阅读者希望更加灵活地表示浏览器窗口以把窗口宽度扩展超过推荐的 194 像素，那么浏览器使用右上角像素来填充扩展的 logo。

2. 频道元素属性

除了必需的 HREF 属性之外，通道元素可以包括 4 个其他属性：

- BASE
- LASTMOD
- PRECACHE
- LEVEL

CHANNEL 标记的一般形式如下：

```
<CHANNEL BASE= "url"   HREF=" url"   LASTMOD=" yyyy-mm-dd"   LEVEL=" n"
  PRECACHE=" yes/no" >
</CHANNEL>
```

（1）BASE 属性是一个 URL，通道中的相对 URL 是与其相关的。例如，如果 BASE 设置为 http://MyWebsite/Mycdf/，那么 HREF 属性可以简单地用 pcbooks.html 代替 http:// MyWebsite / Mycdf /pcbooks.html。子通道可以有自己的 BASE，并且覆盖父通道的 BASE 设置。否则，使用父通道的 BASE。

示例：带有 BASE 属性的 CDF 频道

```
<?xml version="1.0"?>
<CHANNEL BASE="http://metalab.unc.edu/xml/">
  <TITLE>Cafe con Leche</TITLE>
  <ABSTRACT>
    Independent XML news and information for content
    and software developers
  </ABSTRACT>
  <LOGO HREF="cup_ICON.gif" STYLE="ICON"/>
  <LOGO HREF="cup_IMAGE.gif" STYLE="IMAGE"/>
  <LOGO HREF="cup_IMAGE-WIDE.gif" STYLE="IMAGE-WIDE"/>
  <ITEM HREF="books.html">
    <TITLE>Books about XML</TITLE>
    <ABSTRACT>
      A comprehensive list of books about XML
      with capsule reviews and ratings
    </ABSTRACT>
  </ITEM>

  <ITEM HREF="tradeshows.html">
```

```
    <TITLE>Trade shows and conferences about XML</TITLE>
    <ABSTRACT>
      Upcoming conferences and shows with an XML focus
    </ABSTRACT>
  </ITEM>

  <ITEM HREF="mailinglists.html">
    <TITLE>Mailing Lists dedicated to XML</TITLE>
    <ABSTRACT>
      Mailing lists where you can discuss XML
    </ABSTRACT>
  </ITEM>
</CHANNEL>
```

（2）当由 HREF 属性参考的页面经过最新修改时，LASTMOD 属性通过指定日期（以年月日形式如 2000-10-01）告诉阅读者的浏览器是否要求下载。浏览器检测和比较由 Web 服务器提供最后修改日期的 CDF 文件给定的 LASTMOD 日期。当 Web 服务器上的内容改变时，高速缓冲存储器由当前内容修改。阅读者还可以手工刷新通道。

（3）PRECACHE 属性引导浏览器下载通道内容和把这些 Web 页放在阅读者机器的高速缓冲存储器中。当设计通过时，记住有些阅读者几乎是专一性地使用高速缓存方法。因此，通道内容中的任何链接都死掉或者使阅读者能够登录访问它们。如果正在推送文档穿过 Intranet，高速缓存选项并不能做许多事情，就像通过公司在磁盘上复制同样的文件一样。如果正在把内容传送给付费用户，那么可以构造可以放入高速缓存并且可以简单地脱机浏览的内容。另外，嵌入的内容，如声音和 Java applet，不能用内容进行高速缓存。

（4）如果 PRECACHE 属性设置为 yes，那么 LEVEL 属性才有意义。LEVEL 表示当高速缓存内容时，希望浏览器采集多远的距离到达链接层次。这种层次是由文档链接定义的抽象层次，不是由 Web 服务器上文件的目录结构定义的层次。框架页面被看作是作为框架集页面的同一层次的部分，尽管前者要求附加的链接。

### 19.2.2 创建频道

CDF 文档使用所选页的 URL 来确认推送到阅读者的内容。创建频道分为以下三步：

- 确定频道内容。
- 创建频道定义文件。
- 创建从页面到频道的链接。

#### 1. 确定频道内容

在学习复杂的用 CDF 创建频道的技术细节之前，首先必须决定属于频道的内容和这种内容应该如何传送。

当使用已存的站点时，考虑希望出现的层次中每一层的跨度。频道的顶层可能不应该超过 8 层。读者希望使用已经存在的站点结构吗？读者希望从站点中选择指定页，然后用指定频道的层次重新安排这些页吗？

接下来，选择频道内容的传送方法。自动化消息可以使用电子邮件发送给注册阅读者。

可以使用 Microsoft 的通知，它可以加亮在 Explorer 中的激活频道登录或者为阅读者提供浏览图。还可以使用脱机浏览，其中页面内容放在阅读者机器的高速缓冲存储器中，以便以后查看。

选择结构之后，就可以使用全部必要的指令创建实际的 CDF 文件。

2. 创建频道定义文件

CDF 文档包含了描述和确认有关频道内容、调度信息和登录链接频道内容与阅读者桌面或者浏览器的信息，所有这种信息使用特殊的 XML 标志集作标记。这种文档是结构性 XML 文件，可以放在 Web 服务器上以便客户机下载。

CDF 文档以 XML 声明开始，因为 CDF 文档就是 XML 文档且遵循 XML 文档规则。

```
<?xml version=〝1.0〞?>
```

CDF 文档的根目录是<CHANNEL>，该频道元素必须有一个 HREF 属性，它指定被监视修改的页面。根据频道元素通常确认频道中的关键页。例如，下面的频道指向一个每月修改的时事通讯。

```
< CHANNEL HREF=〝http://www.writelivelihood.com/newscover.html〞 >
</CHANNEL>
```

频道元素可以包含其他频道来创建层次，实例如下：

```
<?xml version="1.0"?>
 <CHANNEL HREF="http://www.writelivelihood.com/newscover.html">
 <CHANNEL HREF="http://www.writelivelihood.com/column.html"/>
 </CHANNEL>
 <CHANNEL HREF="http://www.writelivelihood.com/tips.html"/>
 </CHANNEL>
```

该频道定义使用嵌套的频道定义创建一个分层页面。这 2 个页面 column.html 和 tips.html，都位于页面 newscover.html 元素的同一层次。

带有 ITEM 子元素的 CDF 频道：

```
<?xml version="1.0"?>
<CHANNEL HREF="http://metalab.unc.edu/xml/index.html">
  <ITEM HREF="http://metalab.unc.edu/xml/books.html">
  </ITEM>
  <ITEM HREF="http://metalab.unc.edu/xml/tradeshows.html">
  </ITEM>
  <ITEM HREF="http://metalab.unc.edu/xml/mailinglists.html">
  </ITEM>
</CHANNEL>
```

频道内容不局限于一个站点。可以为阅读者包括链接到那些包含相关信息的远程站点上的链接，具体实现如下：

```
<?xml version="1.0"?>
 <CHANNEL HREF="http://www.writelivelihood.com/newscover.html">
       <CHANNEL HREF="http://www.writelivelihood.com/column.html"/>
 </CHANNEL>
       <CHANNEL HREF="http://www.writelivelihood.com/tips.html"/>
</CHANNEL>
       <CHANNEL HREF="http://www.stceo.org/stimulus.htm"/>
```

```
</CHANNEL>
```

在上例中，第二层内容包括了一个外部 Web 站点的参考，这样可以下载另一种杂志。

被动频道（无推送调度的频道）如上例，其实并不能实现很多功能。为了使站点推送更好地工作，需要包括用于修改的调度信息。调度信息可以用于整个频道内的单个项目。

3. 创建从页面到频道的链接

创建频道的第三步是使阅读者可以使用 CDF 文件。为了完成这一步内容，必须创建从 Web 页到 CDF 文件的链接。

基本的选项是使用一个简单的位置参考在页面上增加链接。激活的 Channel 订阅功能扩展了这种选项以便于用登录来确认频道，但是要求一些 JavaScript 来使该进程工作。

链接可以是 HTML 中的简单位置，一般地关闭一些文本或者图像，它要求用户订阅频道，如下面的程序所示。当阅读者在允许使用 CDF 浏览器（Internet Explorer 4.0 及以上版本）中激活该链接时，浏览器下载该文件信息，然后建立客户机和服务器之间的频道连接。其他的浏览器可能要求用户保存该文档。

```
<a href="write_channel.cdf">
    <img   src="channel_subscribe.gif"   width=120   height=120
    border=0   alt="Subscribe to UserAble Write Livelihood's
    Monthly 'zine">
</a>
```

当阅读者激活该链接时，他们把 CDF 文件下载到他们的机器上，然后在他们的机器上创建一个索引。根据他们的选择和 CDF 选项，他们可以回应来自频道的通知，或者手工检查频道。

如果已经增加了到 CDF 的登录，那么这些登录将出现在阅读者机器″上的 2 个位置之一：在桌面上或者在他们的浏览器的频道列表中。

## 19.3 CDF 高级应用

### 19.3.1 CDF 技术进阶

至今为止，有了一个可视连接，阅读者可以用此来快速地访问到站点。但是没有推送内容到阅读者的方法。使用 CHANNEL 属性和 TITLE、ABSTRACT 元素，通过创建一个工作通道可以解决这一问题。为通道增加许多功能可以使其工作更有力。

例如，阅读者可能不知道通道修改——即使他们刷新了他们的通道——除非在 Web 服务器上的 CDF 文件中修改了 LASTMOD 日期。通过调度修改、登录阅读者的访问、确认通道页面和创建用途属性，可以为基本的通道定义增加功能。

1. 调度修改

SCHEDULE 元素指定通道修改的时间。最高层的通道定义是 SCHEDULE 可以拥有的唯一子元素。下面的例子显示了一个通道，在 2000 年 1 月 1 日之前，该通道每 10 天修改一次。

```
<CHANNEL BASE="http://www.writelivelihood.com/">
```

```
    <TITLE>Quality In Development</TITLE>
    <ABSTRACT>TQ100 takes students through the software
  development cycle and shows key documents and
  operations, such as testing.
 </ABSTRACT>
    <LOGO HREF="tune_in_logo_icon.gif" STYLE="ICON"/>
    <LOGO HREF="tune_in_logo_image.jpg" STYLE="IMAGE"/>
    <LOGO HREF="tune_in_logo_image-wide.jpg" STYLE="IMAGE-WIDE"/>
    <SCHEDULE STARTDATE="2000-01-01"   STOPDATE="2000-12-31" TIMEZONE="-0500">
        <INTERVALTIME DAY="10"/>
        <EARLIESTTIME DAY="1" HOUR="0"/>
        <LATESTTIME DAY="2" HOUR="12"/>
    </SCHEDULE>
</CHANNEL>
```

SCHEDULE 元素有 3 个属性：STAUTDATE，STOPDATE 和 TIMEZONE。STARTDATE 表示调度开始的时间，STOPDATE 表示调度结束的时间，目标是通常站点大修理之间的周期。如果以正常的间隔修改 Web 站点结构，那么使用这个间隔需要写一个新的 CDF 来反映新结构。

STARTDATD 和 STOPDATE 使用同样的日期结构：数字年（4）、数字月（2）和数字日（2）（yyyy-mm-dd）。TIMEZONE 属性，那么调度修改根据阅读者的时区发生，而不是根据服务器的时区。

调度最多有 3 个子元素：INTERVAL 是必须的元素，EARLIESTIME 和 LATESTTIME 是可选元素。

上面程序中的调度表示浏览器应该每 10 天修改一次通道（INTERBALTIME DAY=“10”）。该修改应该在每个周期（10 天）第一天半夜（EARLIESTTIME DAY=〝1〞 HOUR=〝0〞）和第二天中午（LATESTTIME DAY=〝2〞 HOUR=〝12〞）之间执行。

修改的最早和最晚时间范围规定服务器加载时间。在这种情况下，如果课程在第二天中午举行，因此给学生留出修改和复习所学内容的时间。

上面的调度使用阅读者的时区来确定修改调度。为了强制根据特定时区修改，在 EARLIESTTIME 和 LATESTTIME 标记中包括可选的 TIMEZONE 属性。

<EARLTESTTIME DAY=〝1〞HOUR=〝0〞 TIMEZONE=〝-0500〞 />

<LATESTTIME DAY=〝2〞HOUR=〝12〞TIMEZONE=〝-0500〞 />

对带有连接局域网的用户，读者可以控制分布和强制系统在服务器的空闲时间内修改。例如，为了推送修改穿过局域网，可以选择星期（例如星期日）和时间（半夜到早晨五点）段。所有浏览器在这 5 小时内修改通道，阅读者必须连接到 Internet 上。

当调度的修改发生时，根据阅读者的订阅选项通知他们。有 2 种类型的修改：改变到 CDF 文件和改变到内容。当在 CDF 文件中的 LASTMOD 改变时，Internet Explorer 增加一个称为光线的通知符号浏览器通道中的通道 logo。虽然这种光线可以或者不可以把修改反映到通道内容中，但是不必每次修改内容时都改变 CDF 文件。

当使用页面上的链接订阅到站点时，阅读者可以选择 3 种改变通知方法，如图 19-1 所示。

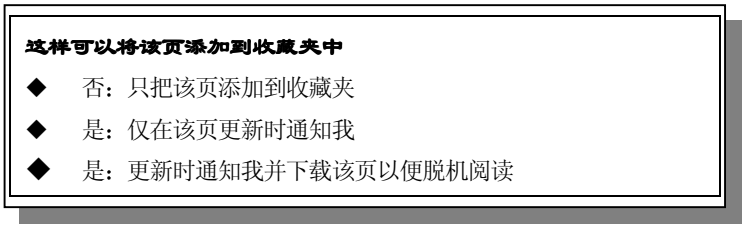


图 19-1

如果阅读者选择第一个选项，那么通道增加到浏览器中并且激活桌面通道栏。然后，阅读者必须选择通道来得到修改。如果阅读者选择第二个选项，那么浏览器执行通道修改检查，通过电子邮件把变化通知给阅读者。如果阅读者选择第三个选项，那么他们将接收到通道内容，并且高速缓存以便脱机查看。

如果使用在 CHANNEL 和 ITEM 标记中的 LEVEL 属性，并设置其值高于零，那么在修改期间浏览器将使 Web 缓慢前进。Web 缓慢前进使浏览器采集到比列在通道中更多的页面。例如，列在通道中的页面包含许多与主题相关的链接。如果站点有一个正好匹配的层次，那么可以安全地增加 LEVEL 属性到最顶端通道标记，并且允许在后面各层中包括全部页。可以从 0 到 3 设置 LEVEL 值。

可以为确认的每个子通道设置层。可以在这些标记中设置前进等层。因此不用列表每个页就能收集整个站点。

如果只想选择特定页，那么，可以为每个页创建一个 CHANNEL 或者 ITEM 定义，并在所有的标记中设置 LEVEL 为零。

当阅读者启动浏览器时，它检查修改调度的通道信息。当找到修改调度时，浏览器修改通道，并且通知阅读者。阅读者可以等待浏览器通道栏中的光线、接收电子邮件或者得到下载到他们机器中的通道内容。

2. 登录阅读者访问

LOG 和 LOGTARGET 元素允许 Web 服务器跟踪阅读者在通道内容中的穿行。该信息保存在阅读者的机器上，所以即使是脱机浏览也可以捕捉到。在通道修改期间，阅读者的机器接收到新的通道内容，并且接收到登录文件，如下例所示：

```
<CHANNEL HREF="index.html" BASE="http://www.writelivelihood.com/">
    <TITLE>Write Livelihood</TITLE>
    <ABSTRACT>Write Livelihood provides technical documentation
  solutions for companies large and small.</ABSTRACT>
    <LOGO HREF="logo_icon.gif" STYLE="ICON"/>
    <LOGO HREF="logo_regular.gif" STYLE="IMAGE"/>
    <LOGO HREF="corp_logo_wide.gif" STYLE="IMAGE-WIDE"/>
    <LOG VALUE="document:view"/>
    <LOGTARGET HREF="c:\work\loghits.txt" METHOD="POST" SCOPE="ALL">
        <PURGETIME HOUR="12"/>
    </LOGTARGET>
    <SCHEDULE STARTDATE="2000-01-01" ENDDATE="2000-12-31" TIMEZONE="-0500">
        <INTERVALTIME DAY="1"/>
```

```
    </SCHEDULE>
</CHANNEL>
```

LOG 元素指定保存在登录文件中的内容。CDF 登录信息可以以扩展文件登格式（EFL）存储。当阅读者每次浏览与 LOG 元素有关的页面时，document:view 输入一个登录。

LOGTARGET 元素决定由阅读者活动生成的登录文件和范围来确认目的。需要一个确认为 HREF 属性，传输方式和范围的目标。HREF 应该确认在服务器上的文件位置，并且包括完整的文件位置。METHOD 属性确认信息如何到达 HREF 位置。POST 是第一个可以使用的 METHOD。最后一个属性：SCOPE 确认要登录的视图。可以使用 OFFLINE，ONLINE 或者 ALL。

登录文件是如何处理是通过通道继承的。因此，LOGTARGET 元素应该作为顶层 CHANNEL 标记的一个子元素出现。登录是非继承的——每一个希望包括在登录中的 CHANNEL 和 ITEM 元素必须包括一个登录元素。

3. 确认通道中的页面

CDF 提供了几种确认通道中内容位置的方法。浏览器首先分析 HREF 属性。然后应付通道内的位置标记。HREF 和位置标记都使用被参考页的完整 URL。

无论使用什么位置链接内容，如果在父通道标记中指定了 BASE 属性，那么就可以在子元素中使用相对 URL。还可以使用 BASE 属性来改变在 Microsoft Internet Explorer 中层次显示的行为。当子元素没有相关页时，在浏览器的窗口中显示基本页。例如，可以如下声明最顶级通道：

```
<CHANNEL BASR=〝http://www.writelivelihood.com/sdc/index.html〝>
```

然后，创建子元素层次：

```
<CHANNEL>
<TILTE>Gathering Information </TITLE>
 <ITEM HREF=" class01.html">
  <TITLE>Product Ideas</TITLE>
```

直到阅读者选择子元素 Product Ideas 时，浏览器窗口中的该页才能改变。

4. 使用 Microsoft Usage 属性

Usage 属性扩展了通道在阅读者桌面上的出现。该 Usage 选项由桌面项、电子邮件、屏幕保护程序和预高速缓存组成。

可以包括一个桌面项，让阅读者将其包括在他们的 Active Desktop（Microsoft Internet Explorer 扩展）中，因而允许把名称和信息放在阅读者监视屏幕上。为了完成该桌面项，需要一个单独的 CDF 文件，包括下面元素：USAGE，OPENAS，HEIGHT，WIDTH 和 CANRESIZE。这些元素是 Microsoft 对基本的 CDF 标准的扩展。它可以或者不必被普遍采纳。

对于 Active Desktop 项，Usage 属性设置为 DesktopComponent。使用 OPENAS 标记，可以表示在 HREF 属性中位置的类型。这可能是一个 HTML 文件或者一幅图像。如果没有使用 OPEANS 标记，那么 Microsoft 假设它是一个 HTML 文件。HEIGHT 和 WIDTH 指定项目要求的桌面上的像素空间。CANRESIZE 标记表示阅读者是否可以改变高度和宽度。

阅读者必须使 Active Desktop 运行以便查看已经包括的任何桌面项。

对于屏幕保护程序选项，用屏幕保护程序创建一个单独的 CDF，它可以是一个 HTML 页，就像顶层通道中的一个项目。如果包括一个屏幕保护程序，该程序可以包括到内容的链接，那么向阅读者提示取代当前的屏幕保护程序。因此，阅读者不能看到本系统的屏幕保护程序。

如果包括有通道用途集到电子邮件的 CDF，那么可以指定一个 HTML 页面是电子邮件内容源。如果不使用电子邮件用途，那么选择接收电子邮件通知的阅读者将收到顶层通道内容。

阅读者可以决定通过电子邮件接收通道修改通知。为了激活电子邮件选项，阅读者必须建立他们的指定电子邮件地址的订阅。下例演示用电子邮件发送通知的频道：

```
<?xml version="1.0"?>
<CHANNEL BASE="http://metalab.unc.edu/xml/">
  <TITLE>Cafe con Leche</TITLE>
  <ABSTRACT>
    Independent XML news and information for content
    and software developers
  </ABSTRACT>
  <LOGO HREF="cup_ICON.gif" STYLE="ICON"/>
  <LOGO HREF="cup_IMAGE.gif" STYLE="IMAGE"/>
  <LOGO HREF="cup_IMAGE-WIDE.gif" STYLE="IMAGE-WIDE"/>

  <ITEM HREF="whatsnews.html">
    <USAGE VALUE="Email"/>
  </ITEM>

  <ITEM HREF="books.html">
    <TITLE>Books about XML</TITLE>
    <ABSTRACT>
      A comprehensive list of books about XML
      with capsule reviews and ratings
    </ABSTRACT>
  </ITEM>

  <ITEM HREF="tradeshows.html">
    <TITLE>Trade shows and conferences about XML</TITLE>
    <ABSTRACT>
      Upcoming conferences and shows with an XML focus
    </ABSTRACT>
  </ITEM>

  <ITEM HREF="mailinglists.html">
    <TITLE>Mailing Lists dedicated to XML</TITLE>
    <ABSTRACT>
      Mailing lists where you can discuss XML
    </ABSTRACT>
  </ITEM>
</CHANNEL>
```

预高速缓存通过内容对于包括那些希望移动到阅读者的机器中由通道页面使用的内容

项，例如声音和图像，是很有用的。通过定义一个包括预高速缓存内容集的通道，可以预高速缓存一项或者一系列项，如下面示例：

```
<CHANNEL>
  <USAGE=" none">
    <ITEM HREF=" http:、、//www.writelivelihood.com/welcome.wav"
      PRECACHE=" yes" >
    <ITME HREF=" http://www.writelihood.com/blowhorn.wav"
     PRECACHE=" yes">
  </USAGE>
</CHANNEL>
```

本示例推送 2 个用在该站点的声音文件，因此避免了与脱机查看有关的困难。

5. 推送软件修改

软件修改通道可以把对软件的修改通知用户和通过 Internet 传送产品。这种传送还可以扩展到包括安装。

为了创建一个软件推送通道，创建一个通道用途设置为 SoftwareUpdate 的 CDF。就像任何其他通道一样，可以包括标题、摘要、图标、Logo 和调度。下面是一个软件修改频道的实例：

```
<?xml version="1.0"?>
<CHANNEL HREF="http://www.whizzywriter.com/updates/2001.html">
  <TITLE>WhizzyWriter 2001 Update</TITLE>
  <ABSTRACT>
    WhizzyWriter 2001 offers the same kitchen sink approach to word processing that WhizzyWriter 2000 was infamous for，but now with tint control! plus many more six-legged friends to delight and amuse!  Don't worry though. All the old arthropods you've learned to love and adore in the last 2000 versions are still here!
  </ABSTRACT>

  <USAGE VALUE="SoftwareUpdate"/>
  <SOFTPKG NAME="WhizzyWriter 2001 with tint control 2.1EA3"
    HREF="http://www.whizzywriter.com/updates/2001.cab"
    VERSION="2001,0,d,3245" STYLE="ActiveSetup">
    <!-- other OSD elements can go here -->
  </SOFTPKG>
</CHANNEL>
```

在软件修改通道内，需要为修改信息指定的软件包定义子元素。<SOFTPKG>标记的属性定义软件包名称、位置、版本、安装选项和样式。

- 软件包名称是用于分布的唯一名称，并且是一个要求的属性。
- 位置使用 HREF 并且指定产品信息驻存的页面的 URL。
- 版本是一个可选属性，用于区分软件的主、次版本。
- 安装选项决定软件名是否应该自动地安装在阅读者的机器上。在 AUTOINSTALL 属性中使用 Yes，不用进一步阅读操作就可以设置通道通过 Internet 把软件包发送到阅读者。
- 样式属性表示 Microsoft 网际组件下载（MSICD）或者 ActiveSetup 方法是否从 Internet

上下载文件。MSICD 选项使用开放软件描述（OSD）指令，ActiveSetup 使用 ActiveX 下载。

- SOFTPKG 元素的子元素包括 ABSTRACT、IMPLEMENTATION、LOGO、TITLE 和 USAGE。
- ABSTRACT 子元素描述软件，并且与标准的 CDF ABSTRACT 使用相同的标记和选项。
- IMPLEMENTATION 元素描述软件包要求的配置。如果在实现标记中描述的要求没有在阅读者机器上找到，那么不能下载和安装。IMPLEMENTATION 元素是一个可选元素，有下列子元素：CODEBASE，LANGUAGE，OS 和 PROCESSOR。
- CODEBASE 元素<CODEBASE FILENAME=“url”HREF=“url” SIZE=“n” STYLE=“ActiveSetup | MSCID”>定义用于分布的可下载文件的位置。
- LANGUAGE 元素使用 RFC 1766 语言代码定义接口的支持语言。如果可以使用多种语言，那么列项之间用分号分开。
- OS 元素可以是 Msc、Win95 或者 Winnt，由此确认软件要求的操作系统。该元素可以有一个子元素 OSVERSION，它确认要求的软件版本。
- PROCESSOR 元素可以是 Alpha、MIPS、PPC 或者 X86。该元素描述软件的机器要求。
- 软件修改通道可以有一个 Logo，与标准的 CDF Logo 一样使用相同的参数：icon、image 和 image-wide。
- 软件修改通道的标题与标准的 CDF 标题使用同样的选项。
- 在软件修改通道中也可以包括 USAGE 标记作为一个子元素。

一旦写好了 CDF 文件，就需要一个软件描述（OSD）文件和一个分布单元。OSD 文件也用 XML 写，并且包含阅读者机器的安装指令。OSD 文件结构与语言在 Microsoft 的下面的 Web 站点描述：http://www.microsoft.com/standards/osd/。分布单元是该软件和 OSD 文件的压缩集合。它使用 CAB 格式压缩，要求一个认证中心的数字记号。

### 19.3.2 综合实例

（1）一个体育站点的频道定义与推送实例：

```
<!DOCTYPE Channel SYSTEM "http://www.w3c.org/Channel.dtd" >

<Channel HREF="http://www.foosports.com/foosports.cdf" IsClonable=YES >

    <IntroUrl
VALUE="http://www.foosports.com/channel-setup.html" />
    <LastMod VALUE="1994.11.05T08:15-0500" />
    <Title VALUE="FooSports" />
    <Abstract VALUE="The latest in sports and atheletics from FooSports" />
    <Author VALUE="FooSports" />

    <Schedule>
      <EndDate VALUE="1994.11.05T08:15-0500" />
      <IntervalTime DAY=1 />
```

```
        <EarliestTime HOUR=12 />
        <LatestTime HOUR=18 />
    </Schedule>

    <Logo HREF=http://www.foosports.com/images/logo.gif    Type="REGULAR" />

    <Item HREF="http://www.foosports.com/articles/a1.html" >
        <LastMod VALUE="1994.11.05T08:15-0500" />

        <Title VALUE="How to get the most out of your mountain
bike" />
        <Abstract VALUE="20 tips on how to work your mountain-bike
to the bone and come out on top." />
        <Author VALUE="FooSports" />
    </Item>

    <Channel IsClonable=NO >
        <LastMod VALUE="1994.11.05T08:15-0500" />
        <Title VALUE="FooSports News" />
        <Abstract VALUE="Up-to-date daily sports news from FooSports" />
        <Author VALUE="FooSports" />

        <Logo HREF=http://www.foosports.com/images/newslogo.gif    Type="REGULAR" />

        <Logo HREF=http://www.foosports.com/images/newslogowide.gif Type="WIDE" />

        <Item
HREF="http://www.foosports.com/articles/news1.html" >
            <LastMod VALUE="1994.11.05T08:15-0500" />
            <Title VALUE="Michael Jordan does it again!"/>
            <Abstract VALUE="Led by Michael Jordan in scoring, the
Chicago Bulls make it to the playoffs again!" />
            <Author VALUE="FooSports" />
        </Item>

        <Item
HREF="http://www.foosports.com/articles/news2.html" />
            <LastMod VALUE="1994.11.05T08:15-0500" />
            <Title VALUE="Islanders winning streak ends"/>
            <Abstract VALUE="The New York islanders' 10-game
winning streak ended with a disappointing loss to the Rangers" />
            <Author VALUE="FooSports" />
        </Item>

    </Channel>
```

```
    <Item
HREF="http://www.foosports.com/animations/scrnsvr.html" />
        <Usage VALUE="ScreenSaver"></Usage>
    </Item>

    <Item
HREF="http://www.foosports.com/ticker.html" />
        <Title VALUE="FooSports News Ticker" />
        <Abstract VALUE="The latest sports headlines from FooSports" />
        <Author VALUE="FooSports" />
        <LastMod VALUE="1994.11.05T08:15-0500"/>

   <!-- This is an example of how Usage can be used for client enhancements -->
        <Usage VALUE="DesktopComponent">
          <Width VALUE=400 />
          <Height VALUE=80 />
        </Usage>

        <Schedule>
          <StartDate VALUE="1994.11.05T08:15-0500" />
          <EndDate VALUE="1994.11.05T08:15-0500" />
          <IntervalTime DAY=1 />
          <EarliestTime HOUR=12 />
          <LatestTime HOUR=18 />
        </Schedule>
    </Item>

</Channel>
```

1．使用 ASP 推送网站的频道

下面的代码生成站点的频道文件，用户可以根据它来建立自己的频道。以后只要用户下载该.cdf 文件，用户就能够订阅自己的站点的频道了。

```
<%
'打开数据库连接，并定义用于格式化的变量
Set DBConn = Server.CreateObject("ADODB.Connection")
DBConn.Open "DSN=YourDSN"

BR = Chr(10)
Quote = Chr(34)

'用来生成频道格式
Body = "<CHANNEL " & BR
Body = Body & "Title = " & Quote & "公司名" & Quote & BR
Body = Body & "LongName = " & Quote & "详细公司名" & Quote & BR
Body = Body & "Abstract = " & Quote & "频道说明" & Quote & BR
'网站生成的频道文件（.cdf）
```

```
Body = Body & "SELF = " & Quote & "http://yoururl/pointcast.cdf" & Quote & BR
Body = Body & "ContentID = " & Quote & "0" & Quote & BR
Body = Body & "Ratings = " & Quote & "(PICS-1.1 "http://www.rsac.org/ratingsv01.html" l gen
        true comment "RSACi North America Server" by "santry@pin-santry.com" for
        "http://yoururl.com" on    "1997.09.18T17:57-0800" r (n 0 s 0 v 0 l 0))" & Quote & BR
Body = Body & "Frequency = " & Quote & "24" & Quote & BR
Body = Body & "Authenticate = " & Quote & "No" & Quote & BR
Body = Body & ">" & BR & BR

SQLQ = "SELECT * FROM YourDB WHERE ENTERDATE = #" & Date() & "#"

'推送更新内容
Set MakeChannel = DBConn.Execute(SQLQ)
Do Until MakeChannel.Eof
Body = Body & "<ITEM " & BR
Body = Body & "Title = " & Quote & MakeChannel("Headline") & Quote & BR
'在本例子中使用数据库，下面的代码指向数据库入口。也就是今天想要显示的文章。
Body = Body & "HREF = " & Quote & "http://yoururl/pointcast-news.asp?Article=" &
MakeChannel("FileName") & Quote & BR
Body = Body & "Type = " & Quote & "HTML" & Quote & BR
Body = Body & "Show = " & Quote & "Channel" & Quote & BR
Body = Body & "Precache = " & Quote & "Yes" & Quote & BR
Body = Body & "Authenticate = " & Quote & "No" & Quote & BR
Body = Body & ">" & BR
Body = Body & "</ITEM>" & BR & BR
MakeChannel.MoveNext
Loop
Body = Body & "</CHANNEL>"
Set fs = CreateObject("Scripting.FileSystemObject")

'生成第一个文件，每次更新时覆盖上一个文件。
Set a = fs.CreateTextFile("d:\yourlocaldrive\pointcast.cdf", True)
a.WriteLine(Body)
a.Close
%>
```

### 19.3.3 CDF 相关工具

CDF 的标记非常庞杂，要记住这么多的标记是很伤脑筋的，而且其规范也是在不断发展的。所以用一些工具来进行设计将大大简化我们的工作（当然，我们不阻止用户使用“写字板”来创建 CDF 文档）。

1．FrontPage 98

FrontPage 98 提供了一个频道定义向导，这样可以很容易地创建一个 CDF 文件，并且指定赞助人所接收的信息。

在 FrontPage 内创建一个 CDF 文档，管理对站点读取的内容，步骤如下：

（1）在 FrontPage Explorer Tools 菜单上，单击 Define Channel，出现频道定义向导。

（2)确定选择了 Create A New Channel Definition Format File For The Current FrontPage Web 选项，然后单击 Next。频道定义向导的第 1 步出现，这时要输入一些信息。

（3）确认主页的路径显示在 Introduction Page 框内。

（4）在 Logo Image 框旁边，单击 Browse 按钮，双击一个 GIF 文件。

（5）在 Logo Image 框旁边，单击 Browse 按钮，双击另一个 GIF 文件，单击 Next。适当的文件名将写在 CDF 文件中，出现第 2 步。

（6）确定 Web URL 在 Source Folder 框中列出。出现第 3 步。此步允许用户从将被送往赞助人的页面的列表中排除页面。

（7）在相应 HTM 文件上点击，按住 Ctrl，单击另一个 HTM 文件。单击 Exclude，然后单击“Next”。出现第 4 步。在此步中，可以修正包括在 CDF 文件中的页面属性。

（8）确定在 Channel Item 列表中选择了某个 HTM 文件。在 Abstract 框内置插入点，然后键入页面名称。

（9）确定选择了 Specify Usage 选项和 Channel 控制框，单击 Next。出现第 5 步。在此步中，可以指定数据，向 Web 站点通知赞助人的更新。

（10）再次单击 Next。出现第 6 步。可以不提供一个 URL 来记录赞助人信息。

（11）单击 Next，发展到第 7 步。

（12）在 Addtional Option 区域内，确定选择了 Place A Button 制框，并且确定反映新 CDF 文件的“File Name”框被保存在 Web 站点。

（13）单击 Save。CDF 文件被保存在 Web 站点。赞助人能够看到主页和页面信息，并且使用这种搜索表单发现其它的信息。

2. 频道生成大师 Channel Maker

Channel Maker 是 Any Ware 公司的产品，目前最新的版本是 1.01。使用 Channel Maker，用户不需要记一条 XML 语句，所需要做的只是怎样来让 Channel Maker 生成满意的频道。

一个最基本的完整频道包括四个部分：一个 CDF 文件和三个图形标识文件。第一个图形文件是图标文件 ICO，格式为 ICO 或 GIF，大小为 16×16 个像素，用于显示在 IE 的收藏菜单下的频道子菜单中；第二个图形文件是 Image，格式为 JPEG 或 GIF，大小为 80×32 个像素，用于显示在桌面频道条上；第三个图形文件是 Wide Image，格式为 GIF 或 JPEG，大小为 194×32 个像素，用于显示在 IE 浏览器中的频道栏中。如果 Wide Image 不存在，浏览器将使用 Image 来代替。

复杂的频道可以包含子频道和项目。子频道可以使用户有层次地组织自己的 Web 页面，针对不同的子频道和项目做不同的设置。运行 Channel Maker，一般它会自动开始一个新的频道文件。如果已经调入了一个频道文件，可在 File 菜单中选择 New Channel File，然后该频道文件缺省的第一级频道就是 New Channel，当然用户也可以任意改变频道的标题。

3. 菜单选项说明:

- File Reopen 是对 Open 功能的增强，它可以打开最近打开过的频道文件。
- File Test Channel File 是对生成的频道文件进行测试，选择了这一项后，会运行 IE 显

示当前制作的频道。一般最好先运行一个 Web 服务器程序如 O′ Reilly 的 WebSite 等。如果没有预定该频道，IE 就会询问是只加入到频道栏里、仅在更新时通知或更新时通知并下载。如果已经预定了该频道，IE 只是简单地显示该频道。

- File Up date All Last Modified Dates 用于更新所有项目和子频道的最后更改日期。
- File Check All Links 检查所有的频道页面、文件连接、项目等是否存在。如果工作在离线状态下，Channel Maker 只检查本地的拷贝是否存在，不过最好还是运行 Web 服务器程序检查，以确保万无一失。
- File Properties 设置当前频道文件的属性。XML Version 的缺省设置为 1.0，不需要改变。XML Encoding 则根据不同的需要做不同的设置：如果是英文频道（用于对国外发布），编码方法使用 Windows－1252；如果是中文频道文件（对国内发布），则使用 gb2312。
- File Preferences 对 Channel Maker 进行设置。Automatically Check Links Before Saving 是在保存频道时先对频道进行检查；Automatically Update All Last Modified Dates Before Saving 是在保存频道文件时更新频道所有的最后更改日期；Use HTTP Proxy 是设置代理服务器，如果是通过 HTTP 代理服务器访问自己的 WWW 服务器，则需要选定这一项，并填入代理服务器的地址和端口。在页面映射页里面，可以设置本地路径对 WWW 服务器的映射，使得 Channel Maker 在离线方式下能够正常工作；在 Filename used for default page 选择 WWW 服务器缺省的主页，Channel Maker 提供了 5 个选择（index.htm、index.html、defaul.htm、defaul.html、defaul.asp），如果不在这 5 个里面就在文本框填入文件名，例如 O′ Reilly Web Site1.1 的缺省主页为 index.html－ssi。
- Channel New Sub－Channel 创建一个新的子菜单。
- Channel New Item 添加一条新的项目。
- Channel Delete 删除子菜单或项目。

如果在 Address 栏里输入一个 URL 地址，Channel Maker 会调入 IE 连接到该地址浏览。

在 Address 栏下选择 Channel 页，URL of this channel′ s cover page 就是右边频道选定的子频道或项目对应的超文本连接；Base URL of this channel（项目没有这一选项）则是频道的关联地址。选择 General 页，在 Enter the Title of the Channel or Item 下面填入频道或项目的标题；在 Enter an abstract for the Channel or Item 下面填入该频道或项目的简要介绍。再选择 Logos 页，在 IconURL 中填入频道图标文件的路径和文件名，也可以用文本框右边的 Find 按钮查找该文件；ImageURL 和 WideImageURL 类似。选择 File Save，第一个频道文件就生成了。注意，最好把文件存在 WWW 服务器的主页所在的目录。再选择 File Test Channel File，为频道起一个名字，选择一种加入频道的方式，频道就会同时显示在 IE 的频道栏和活动桌面的频道条里了。

仅仅制作了频道还是不够的，还需要发布它，也就是说让别的人能够预定。这也很简单，只需要在主页加入一个链接。如：

〈AHREF="http://www.mycompany.com/mychannel.cdf"〉预定频道！〈/A〉

为了引人注目，利用图形将是更好的办法。将“预定频道”的 GIF 文件放在 WWW 网址对应的目录中，用 JavaScript 将频道加入用户的活动频道条，具体如下：

〈AHREF="javascript:external.addChannel

（′ http://www.mycompany.com/mychannel.cdf′ ）；"〉

〈IMG SRC="add-channel.gif" WIDTH=138HEIGHT=24ALT="请预定我们的频道！" BORDER=0〉 〈/A〉

当然，也可以以 Email 的方式提供给别人，在 Email 中加入类似以下的链接：

http://www.mycompany.com/mychannel.cdf

如果想让更多的人预定，可以将频道放在微软公司的主页上，具体信息可参看：

http://www.microsoft.com/ie-intl/cn/ie40/download/cdf /iechannl.htm。

从 http://www.anyware.co.uk 下载的 Channel Maker1.01 是共享版，它的功能受到一定的限制，如不能在频道中加入子频道等。为了使用 Channel Maker 的所有功能，需要给 AnyWare 发一封 Email，告诉它用户名，AnyWare 将分发一个序列号。拿到该序列号后，运行 Channel Maker，选择 Help 菜单下的 Register，输入用户名和序列号，就可以使用 Channel Maker 的所有功能了，只不过 30 天的试用期还没有改变。

## 19.4 小　　结

通过本章的学习，我们对 CDF 技术有了全面的认识和了解，我们可以利用灵活的频道来在客户端浏览器和 Web 服务器之间建立起虚拟链接，自动发送数据和信息。这种 Web 上的“推拉”过程和技术使得网络信息的发布更为轻松和有序，CDF 与其他技术（如：OSD，开发软件描述）相结合，将给未来的 Web 开发注入新的活力。

# 第二十章　Web 站点的设计实现

在这部分的最后一章里，我们综合前面所讨论和介绍的各种相关技术实现一个基于 XML 的 Web 站点。此 Web 站点需要 IIS（Internet Information Server 网络信息服务）的支持，所以必须运行 Windows NT 下的 IIS 或者个人网络服务（Personal Web Server）并安装动态服务页面 ASP（Active Service Pages）。同时，服务器还要支持 XML 组件（如：DOM 等）。

本章包括以下内容：

- 站点整体设计
- 站点创建

## 20.1　站点整体设计

在进行站点整体设计之前，对设计构思和布局结构进行验证是必要的。验证构思有助于更好地设计和建立站点的用户界面；验证布局结构有助于选择实现站点的外观。实践表明，我们可以按照这种结构来设计表现站点的 XML 文档。

我们的这个 Web 站点主要是基于一份包括文章、新闻、产品浏览的在线杂志。此站点也包括一个数据库，用户可以用它来查找 XML 的相关开发工具。站点结构如图 20-1。

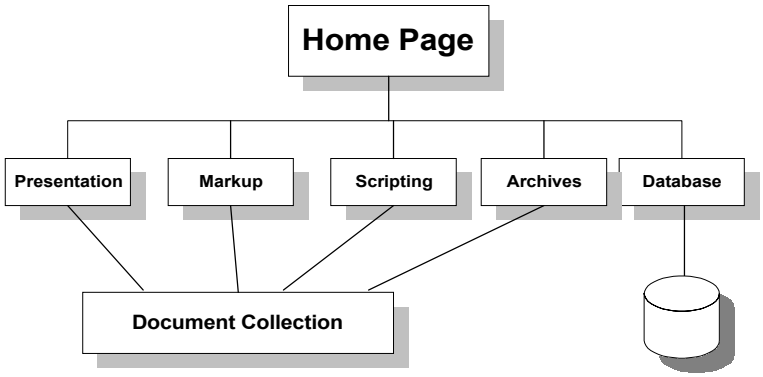


图 20-1

为使站点简单易用，我们为访问的用户提供了几个他们可能感兴趣的主题。Presentation 域提供与用户界面设计、风格表单实现等相关的文档；Markup 域包括表现语言（HTML 和 XML）；Scripting 域讨论使用 JavaScript、Perl、Python 和 VBScript 的技巧；Arhives 域包含数据库中存储的可供查询的 XML 工具数据的背景和相关信息。

与此相似，目录结构也相应减化，提供两个目录层次：根目录和一级子目录，这对于表现整体结构已经足够，如图 20-2。用户可以按照自己的想法扩展目录结构。

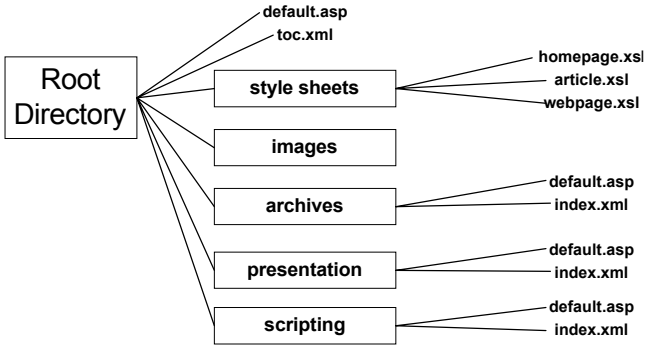


图 20-2

如上图，根目录下有两个文件：default.asp 和 toc.xml。default.asp 是用户访问站点时的服务文件；toc.xml 是表现主页信息的 XML 文档。主页的风格表单是 homepage.xsl。各子目录中同样包括两个文件，一个是 default.asp，完成浏览器类型检测的动态服务页面文件；另一个是 index.xml，表现子目录索引页面的 XML 文档。

## 20.2 站 点 创 建

接下来我们要考虑的是页面如何提供服务：是在服务器端实现还是在浏览器中执行？我们用 default.asp 来检测用户的浏览器类型，如果是 IE5 的话，服务页面会将用户定向到 XML 页面上去，因为页面引用了风格表单，所以浏览器可以收集相关资源，并在客户端处理 XML 文档。如果不是 IE5，那么服务页面在服务器端处理文档。下面是 default.asp 的源代码：

```
<%@ LANGUAGE=JScript %>
<% Response.Buffer = "True"; %>
    <%
    var testAgent = Request.ServerVariables("HTTP_USER_AGENT");
    var checkBrowser = Server.CreateObject("MSWC.BrowserType");

    if ( checkBrowser.version == 5.0)
        Response.Redirect("toc.xml");

    else {
        Response.Write("This is an XML page and requires Internet Explorer 5 to view. We will have support for all browsers up in a few
days. We apologize for the inconvenience.");
        var xmlDocument =
            Server.CreateObject("Microsoft.XMLDOM");
        var xslDocument =
            Server.CreateObject("Microsoft.XMLDOM");

        xmlDocument.load(Server.MapPath("toc.xml"));
        xslDocument.load(Server.MapPath("stylesheets/homepage.xsl"));
```

```
        Response.Write(xmlDocument.transformNode(xslDocument));
    }

    %>
</BODY>
</HTML>
```

1. Toc.xml

因为风格表单（*.xsl）可以表现导航条、链接图标等页面元素，所以 XML 文档可以只关注于主页面信息的存放和变化，不需考虑页面布局和表现形式的 XML 文档很简单，下面是 toc.xml 的代码：

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="http://www.beyondhtml.com/stylesheets/homepage.xsl"?>
<homepage>
        <title>BeyondHTML.com Home Page</title>
        <H1>Contents</H1>
        <pubDate>September, 1999</pubDate>
        <contents>
                <headline href="news.xml">
            News&amp;amp;Views
        </headline>
                <abstract>
        <para>
        The W3C releases the first public working draft of the Scalable Vector Graphics (SVG) format.
        </para>
                </abstract>
                <headline2 href="workbench.xml">Building an XML Workbench</headline2>
                <abstract2>
                <para>This month, Michael shows how you can set up a complete workbench of tools to process XML on your Web server. The workbench includes an XML and XSL processor, and a Java Servlet that delivers XML  documents. The best news is that the tools are freely available from IBM and Sun. </para>
                </abstract2>
                <headline3 href="frames.xml">Frames With Style</headline3>
                <abstract3>
                <para>Because of excessive abuse, frames have gotten a bad "rep" from Web designers and developers. But used properly, frames can enhance navigation while easing maintenance of your site. Michael shows you how.</para>
        </abstract3>
        </contents>
        <notes>Interested in a topic? Drop us a line at editors@beyondHTML.com</notes>
</homepage>
```

2. Homepage.xsl 文件的创建

这里我们来看看主页面的风格表单，首先它为主页面生成级联表规则，然后创建文档的<body>。需要注意的是超链接的创建，举例来说，主页面在内容表中引用三篇小说，每篇小

说都有用级联表规则约定的题头和内容摘要。如果用户想要阅读小说，他就点击题头，题头链接到实际的小说文本。在 XML 文档中，<headline>元素用 href 属性指定一个 URL；在样式表单中，必须创建一个锚标记使得题头成为可链接的，通常可以这样写：

```
<A href=“news.xml”>Today’s News
</A>
```

在 XSL 样式表单中，可以用 headline[@href]来获取属性值，用 article/headline 来获取锚元素的内容。所以可以这样设计：

```
<A href=“
    <xsl:value-of select=“headline[@href]”/>
 >
    <xsl:value-of select=“article/headline”/>
 </A>
```

下面是完整的风格表单 homepage.xsl 的代码：

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

<xsl:stylesheet
    xmlns:xsl="http://www.w3.org/TR/WD-xsl"
    xmlns="http://www.w3.org/TR/REC-html40"
    result-ns="">

  <!-- Root template -->
  <xsl:template match="/">
    <HTML>
      <HEAD>
        <META http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>

         <!-- Navigation Bar Styles -->

        <STYLE TYPE="text/css">
         A:link
             COLOR:    Navy;
             text-decoration: none }
         A:visited { COLOR: Navy }
         A:active { COLOR: blue }

         #NavText
             font-weight: bold;
             font-size: 14px;
             text-decoration: none;
             font-family: "Times New Roman", "Garamond", "serif" }

         #NavHead
             font-weight: bold;
             text-align: center;
             font-size: 24px;
```

```
    text-decoration: none;
    font-family: "Times New Roman", "serif", "Garamond"; color: white;
    display: block; background-color: navy; border-style: outset;
    margin-left: 2%; margin-right: 5% }

<!-- General Styles -->
H1 {
    color: #FF0000;
      background-color: #FFFFFF;
        text-transform: Capitalize;
        text-align: Left; }


<!-- Document Styles -->

.headline {
    color: #FF0000;
    background-color: #FFFFFF;
    text-transform: Capitalize;
    text-align: Left; }

.deck {
    font-style: italic;
    font-size: 14px;
    font-weight: bold;
    color: black;
    margin-left: 64px;
    font-family: Arial, helvetica, sans-serif;}

.byline {
    color: Navy;
    font-weight: bold;
    font-size: 14px;
    font-family: Arial, helvetica, sans-serif;}

.pubDate {
    color: Red;
    font-weight: normal;
    font-size: 12px;
    font-family: Arial, helvetica, sans-serif;}

.copyright {
    color: Red;
    font-weight: normal;
    font-size: 12px;
    font-family: Arial, helvetica, sans-serif;}
```

```
.aBody { display: block;
    font-weight: normal;
    font-size: 12px;
    font-family: "Arial", "Garamond", "serif"; }

.dropCap { background: white;
    color: red;
    float: left;
    vertical-align: text-top;
    font-size: 24px;
    font-style: bold;
    border: none; }

.bold { font-style: bold; }
.ital { font-style: italic; }

#BoxCopy { color: white;
    background-color: red;
    vertical-align: text-bottom;
    font-size: 24px;
    font-style: bold;
    font-family: "Times New Roman", "serif", "Garamond"; color: white;
    padding-left: 1px;
    padding-right: 3px;
    text-decoration: none;
    border: none; }

#BoxCopy1 { color: white;
     background-color: gray;
     vertical-align: text-top;
     font-size: 24px;
     font-style: bold;
     text-decoration: none;
     border: inset }

#BoxCopy2 { color: navy;
     background-color: white;
     float: right;
     vertical-align: text-top;
     font-size: 20px;
     font-style: bold;
     border: none; }
</STYLE>
<TITLE>
  <xsl:value-of select="article/headline"/>
```

```
  </TITLE>

 </HEAD>
 <BODY>

<TABLE WIDTH="100%">
  <TR>
    <TD WIDTH="121" HEIGHT="211" BACKGROUND="/images/navBackground.jpg" VALIGN="TOP">

    <Span id="NavHead">InSite</Span>

    <P>
    <A HREF="present/default.asp" TARGET="_top">
        <SPAN id="NavText">Presentation</SPAN>
    </A></P>

    <P>
    <A HREF="markup/default.asp" TARGET="_top">
        <SPAN id="NavText">Markup</SPAN>
    </A></P>

    <P>
    <A HREF="markup/xml/default.asp" TARGET="MainPane">
        <SPAN id="NavText">XML</SPAN>
    </A></P>

    <P>
    <A HREF="scripting/default.asp" TARGET="_top">
        <SPAN id="NavText">Scripting</SPAN>
    </A></P>

    <P>
    <A HREF="db/searchform.html" TARGET="_top">
        <SPAN id="NavText">Tools Database</SPAN>
    </A></P>

    <P>
    <A HREF="archives/default.asp" TARGET="_top">
        <SPAN id="NavText">Archives</SPAN>
    </A></P>

    <P>
    <A HREF="bio.shtml"><SPAN id="NavText">About Us</SPAN>
    </A></P>

    <P>
```

```
        <A HREF="http://www.beyondhtml.com/" TARGET="_top">
            <SPAN id="NavText">Home</SPAN>
        </A></P></TD>

        <TD WIDTH="35" HEIGHT="211" BGCOLOR="#FFFFFF" VALIGN="TOP"></TD>
        <TD WIDTH="442" HEIGHT="211" BGCOLOR="#FFFFFF" VALIGN="TOP">

    <MAP NAME="logosmall">
        <AREA SHAPE="RECT" COORDS="0,1,199,52"
                HREF="http://www.beyondhtml.com"
                ALT="Jump to Home page"></AREA>
        <AREA SHAPE="default" HREF="http://www.beyondhtml.com"></AREA>
    </MAP>

    <P>
      <BR></BR> <IMG SRC="/images/logo.gif"
                    ALT="BeyondHTML Logo" ALIGN="MIDDLE"
                    WIDTH="426" HEIGHT="112">
                </IMG>
    </P>
<!-- template rules go here -->

    <H1>
        <xsl:value-of select="homepage/H1"/>
    </H1>
        <xsl:value-of select="homepage/pubDate"/>
    <HR></HR>

    <!-- Create an Anchor element and a link to the headline -->
    <xsl:element name="A">
        <xsl:attribute name="href">
          <xsl:apply-templates select="homepage/contents/headline[@href]"/>
        </xsl:attribute >
    </xsl:element>
                <Span ID="BoxCopy2">
                    <xsl:value-of select="homepage/contents/headline"/>
                    <xsl:apply-templates/>
                </Span><BR></BR>
    <xsl:element name="/A">
      <xsl:apply-templates />
    </xsl:element>

    <DIV Class="aBody">
      <P><xsl:value-of select="homepage/contents/abstract/para"/></P>

    <!-- Create an Anchor element and a link to headline2 -->
```

```
  <xsl:element name="A">
    <xsl:attribute name="href">
      <xsl:apply-templates select="homepage/contents/headline2[@href]"/>
    </xsl:attribute >
  </xsl:element>
                  <Span ID="BoxCopy">
                    <xsl:value-of select="homepage/contents/headline2"/>
                  </Span><BR></BR>
  <xsl:element name="/A">
    <xsl:apply-templates />
</xsl:element>

                  <P><xsl:value-of select="homepage/contents/abstract2/para"/></P>

  <!-- Create an Anchor element and a link to headline3 -->
  <xsl:element name="A">
    <xsl:attribute name="href">
      <xsl:apply-templates select="homepage/contents/headline3[@href]"/>
    </xsl:attribute >
  </xsl:element>
                  <Span ID="BoxCopy1">
                    <xsl:value-of select="homepage/contents/headline3"/>
                  </Span><BR></BR>
<xsl:element name="/A">
    <xsl:apply-templates />
</xsl:element>

                  <P><xsl:value-of select="homepage/contents/abstract3/para"/></P>
                  <P><I><xsl:value-of select="homepage/notes"/></I></P>
              </DIV>
      </TD></TR>
    </TABLE>

    <P ALIGN="CENTER"><FONT SIZE="-1"> [<A HREF="/present/index.xml">Presentation</A>]
    [<A HREF="/markup/index.xml">Markup</A>] [<A HREF="/markup/xml/index.xml">XML</A>]
    <A HREF="/scripting/index.xml">[Scripting</A>] [<A HREF="/db/searchform.html">Tools
      Database</A>] [<A HREF="/archives/index.xml">Archives</A>] [<A HREF="/bio.shtml">About
      Us</A>] [<A HREF="http://www.beyondhtml.com/">Home</A>]</FONT> </P>
  <HR></HR>

  <P><FONT SIZE="-1">Copyright (c) 1998, 1999.
    <A HREF="mailto:mfloyd@beyondhtml.com">Michael Floyd</A> . All Rights Reserved</FONT>
  </P>

  </BODY>
  </HTML>
```

```
</xsl:template>

<xsl:template match="headline[@href]">
    <xsl:value-of select="@href"/>
</xsl:template>

<xsl:template match="headline2[@href]">
    <xsl:value-of select="@href"/>
</xsl:template>

<xsl:template match="headline3[@href]">
    <xsl:value-of select="@href"/>
</xsl:template>

<xsl:template match="aBody">
    <P>
        <xsl:apply-templates/>
    </P>
</xsl:template>

<xsl:template match="dropCap">
    <DIV Class="dropCap">
        <xsl:apply-templates />
    </DIV>
</xsl:template>

<xsl:template match="bold">
    <B>
        <xsl:apply-templates/>
    </B>
</xsl:template>

<xsl:template match="italic">
    <I>
        <xsl:apply-templates/>
    </I>
</xsl:template>

<xsl:template match="byline[@Email]">
    <A HREF="mailto:mfloyd@BeyondHTML.com">
        <xsl:apply-templates/>
     </A>
</xsl:template>


<xsl:template match="text()">
```

```
        <xsl:value-of />
    </xsl:template>
  <!-- template rules end here -->


</xsl:stylesheet>
```

其他相关的文档代码：

**article.xsl**

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

<xsl:stylesheet
    xmlns:xsl="http://www.w3.org/TR/WD-xsl"
    xmlns="http://www.w3.org/TR/REC-html40"
    result-ns="">

  <!-- Root template -->
  <xsl:template match="/">
    <HTML>
     <HEAD>
       <META http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
       <STYLE TYPE="text/css">

        <!-- Navigation Bar Styles -->

        A:link
            COLOR:    Navy;
            text-decoration: none }
        A:visited { COLOR: Navy }
        A:active { COLOR: blue }

        #NavText
            font-weight: bold;
            font-size: 14px;
            text-decoration: none;
            font-family: "Times New Roman", "Garamond", "serif" }

        #NavHead
            font-weight: bold;
            text-align: center;
            font-size: 24px;
            text-decoration: none;
            font-family: "Times New Roman", "serif", "Garamond"; color: white;
            display: block; background-color: navy; border-style: outset;
            margin-left: 2%; margin-right: 5% }

        <!-- Document Styles -->
```

```
.headline {
    color: #FF0000;
    background-color: #FFFFFF;
    text-transform: Capitalize;
    text-align: Left; }

.deck {
    font-style: italic;
    font-size: 14px;
    font-weight: bold;
    color: black;
    margin-left: 64px;
    font-family: Arial, helvetica, sans-serif;}

.byline {
    color: Navy;
    font-weight: bold;
    font-size: 14px;
    font-family: Arial, helvetica, sans-serif;}

.pubDate {
    color: Red;
    font-weight: normal;
    font-size: 12px;
    font-family: Arial, helvetica, sans-serif;}

.copyright {
    color: Red;
    font-weight: normal;
    font-size: 12px;
    font-family: Arial, helvetica, sans-serif;}

.aBody { display: block;
    font-weight: normal;
    font-size: 12px;
    font-family: "Arial", "Garamond", "serif"; }

.dropCap { background: white;
    color: red;
    float: left;
    vertical-align: text-top;
    font-size: 24px;
    font-style: bold;
    border: none; }
```

```
        .bold { font-style: bold; }
        .ital { font-style: italic; }

        #BoxCopy { color: white;
            background-color: red;
            vertical-align: text-bottom;
            font-size: 24px;
            font-style: bold;
            font-family: "Times New Roman", "serif", "Garamond"; color: white;
            padding-left: 1px;
            padding-right: 3px;
            text-decoration: none;
            border: none; }

        #BoxCopy1 { color: white;
             background-color: gray;
             vertical-align: text-top;
             font-size: 24px;
             font-style: bold;
             text-decoration: none;
             border: inset }

        .BoxCopy2 { color: navy;
             background-color: white;
             float: right;
             vertical-align: text-top;
             font-size: 20px;
             font-style: bold;
             border: none; }
    </STYLE>
    <TITLE>
       <xsl:value-of select="article/headline"/>
    </TITLE>

  </HEAD>
  <BODY>

<TABLE WIDTH="100%">
    <TR>
       <TD WIDTH="121" HEIGHT="211" BACKGROUND="images/navBackground.jpg" VALIGN="TOP">

       <Span id="NavHead">InSite</Span>

       <P>
       <A HREF="/present/default.asp" TARGET="_top">
            <SPAN id="NavText">Presentation</SPAN>
```

```
    </A></P>

    <P>
    <A HREF="/markup/default.asp" TARGET="_top">
        <SPAN id="NavText">Markup</SPAN>
    </A></P>

    <P>
    <A HREF="/markup/xml/default.asp" TARGET="MainPane">
        <SPAN id="NavText">XML</SPAN>
    </A></P>

    <P>
    <A HREF="/scripting/default.asp" TARGET="_top">
        <SPAN id="NavText">Scripting</SPAN>
    </A></P>

    <P>
    <A HREF="/db/searchform.html" TARGET="_top">
        <SPAN id="NavText">Tools Database</SPAN>
    </A></P>

    <P>
    <A HREF="/archives/default.asp" TARGET="_top">
        <SPAN id="NavText">Archives</SPAN>
    </A></P>

    <P>
    <A HREF="bio.shtml"><SPAN id="NavText">About Us</SPAN>
    </A></P>

    <P>
    <A HREF="http://www.beyondhtml.com/" TARGET="_top">
        <SPAN id="NavText">Home</SPAN>
    </A></P></TD>

    <TD WIDTH="35" HEIGHT="211" BGCOLOR="#FFFFFF" VALIGN="TOP"></TD>
    <TD WIDTH="442" HEIGHT="211" BGCOLOR="#FFFFFF" VALIGN="TOP">

<MAP NAME="logosmall">
    <AREA SHAPE="RECT" COORDS="0,1,199,52"
          HREF="http://www.beyondhtml.com"
          ALT="Jump to Home page"></AREA>
    <AREA SHAPE="default" HREF="http://www.beyondhtml.com"></AREA>
</MAP>
```

```
    <P>
      <BR></BR> <IMG SRC="images/logo.gif"
                     ALT="BeyondHTML Logo" ALIGN="MIDDLE"
                     WIDTH="426" HEIGHT="112">
                </IMG>
    </P>
<!-- template rules go here -->

                <Span ID="BoxCopy">
                    <xsl:value-of select="article/headline"/>
                </Span><BR></BR>

                <DIV Class="deck">
                    <xsl:value-of select="article/deck"/>
                </DIV>
                <BR></BR>

                <DIV Class="byline">
                    By
                    <xsl:value-of select="article/byline"/>
                </DIV>
                <BR></BR>
                <DIV Class="aBody">
                    <P><xsl:value-of select="article//aBody/para1"/></P>
                    <P><xsl:value-of select="article//aBody/para"/></P>
                    <P><xsl:value-of select="article//aBody/para2"/></P>
                </DIV>
          </TD></TR>
      </TABLE>

      <P ALIGN="CENTER"><FONT SIZE="-1"> [<A HREF="/present/index.xml">Presentation</A>]
      [<A HREF="/markup/index.xml">Markup</A>] [<A HREF="/markup/xml/index.xml">XML</A>]
      <A HREF="/scripting/index.xml">[Scripting</A>] [<A HREF="/db/searchform.html">Tools
        Database</A>] [<A HREF="/archives/index.xml">Archives</A>] [<A HREF="/bio.shtml">About
        Us</A>] [<A HREF="http://www.beyondhtml.com/">Home</A>]</FONT> </P>
    <HR></HR>

    <P><FONT SIZE="-1">Copyright (c) 1998, 1999.
      <A HREF="mailto:mfloyd@beyondhtml.com">Michael Floyd</A> . All Rights Reserved</FONT>
    </P>

    <P><FONT SIZE="-1"><I>Last Modified:

      <!--#echo var="LAST_MODIFIED"--></I></FONT></P>

      </BODY>
```

```
    </HTML>
  </xsl:template>

  <xsl:template match="aBody">
      <P>
          <xsl:apply-templates/>
      </P>
  </xsl:template>

  <xsl:template match="dropCap">
      <DIV Class="dropCap">
          <xsl:apply-templates />
      </DIV>
  </xsl:template>

  <xsl:template match="bold">
      <B>
          <xsl:apply-templates/>
      </B>
  </xsl:template>

  <xsl:template match="italic">
      <I>
          <xsl:apply-templates/>
      </I>
  </xsl:template>

  <xsl:template match="byline[@Email]">
      <A HREF="mailto:mfloyd@BeyondHTML.com">
          <xsl:apply-templates/>
       </A>
  </xsl:template>

 <!-- template rules end here -->


</xsl:stylesheet>
```

## index.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="http://www.beyondhtml.com/stylesheets/Webpage.xsl"?>
<article>
        <headline>Scripting Languages</headline>
        <deck>Add late breaking messages here...</deck>
        <pubDate>Last Modified:</pubDate>
        <aBody>
```

```
            <para1>
                <dropCap>W</dropCap>elcome to the Scripting languages Home page. This
            template is just a place holder for your XML page. Simply fill ot the
            template to create your own custom site based on XML.
        </para1>
        </aBody>
        <copyright>Copyright (c) 1999, Michael Floyd. All Rights Reserved</copyright>
</article>
```

## Webpage.xsl

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

<xsl:stylesheet
    xmlns:xsl="http://www.w3.org/TR/WD-xsl"
    xmlns="http://www.w3.org/TR/REC-html40"
    result-ns="">

  <!-- Root template -->
  <xsl:template match="/">
    <HTML>
     <HEAD>
       <META http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
       <STYLE TYPE="text/css">

        <!-- Navigation Bar Styles -->

        A:link
            COLOR:    Navy;
            text-decoration: none }
        A:visited { COLOR: Navy }
        A:active { COLOR: blue }

        #NavText
            font-weight: bold;
            font-size: 14px;
            text-decoration: none;
            font-family: "Times New Roman", "Garamond", "serif" }

        #NavHead
            font-weight: bold;
            text-align: center;
            font-size: 24px;
            text-decoration: none;
            font-family: "Times New Roman", "serif", "Garamond"; color: white;
            display: block; background-color: navy; border-style: outset;
            margin-left: 2%; margin-right: 5% }
```

```
<!-- Document Styles -->

.headline {
    color: #FF0000;
    background-color: #FFFFFF;
    text-transform: Capitalize;
    text-align: Left; }

.deck {
    font-style: italic;
    font-size: 14px;
    font-weight: bold;
    color: black;
    margin-left: 64px;
    font-family: Arial, helvetica, sans-serif;}

.byline {
    color: Navy;
    font-weight: bold;
    font-size: 14px;
    font-family: Arial, helvetica, sans-serif;}

.pubDate {
    color: Red;
    font-weight: normal;
    font-size: 12px;
    font-family: Arial, helvetica, sans-serif;}

.copyright {
    color: Red;
    font-weight: normal;
    font-size: 12px;
    font-family: Arial, helvetica, sans-serif;}

.aBody { display: block;
    font-weight: normal;
    font-size: 12px;
    font-family: "Arial", "Garamond", "serif"; }

.dropCap { background: white;
    color: red;
    float: left;
    vertical-align: text-top;
    font-size: 24px;
    font-style: bold;
```

```
            border: none; }

        .bold { font-style: bold; }
        .ital { font-style: italic; }

        #BoxCopy { color: white;
            background-color: red;
            vertical-align: text-bottom;
            font-size: 24px;
            font-style: bold;
            font-family: "Times New Roman", "serif", "Garamond"; color: white;
            padding-left: 1px;
            padding-right: 3px;
            text-decoration: none;
            border: none; }

        #BoxCopy1 { color: white;
                background-color: gray;
                vertical-align: text-top;
                font-size: 24px;
                font-style: bold;
                text-decoration: none;
                border: inset }

        .BoxCopy2 { color: navy;
                background-color: white;
                float: right;
                vertical-align: text-top;
                font-size: 20px;
                font-style: bold;
                border: none; }
    </STYLE>
    <TITLE>
       <xsl:value-of select="article/headline"/>
    </TITLE>

  </HEAD>
  <BODY>

<TABLE WIDTH="100%">
    <TR>
       <TD WIDTH="121" HEIGHT="211" BACKGROUND="/images/navBackground.jpg" VALIGN="TOP">

       <Span id="NavHead">InSite</Span>

       <P>
```

```
<A HREF="/present/default.asp" TARGET="_top">
    <SPAN id="NavText">Presentation</SPAN>
</A></P>

<P>
<A HREF="/markup/default.asp" TARGET="_top">
    <SPAN id="NavText">Markup</SPAN>
</A></P>

<P>
<A HREF="/markup/xml/default.asp" TARGET="MainPane">
    <SPAN id="NavText">XML</SPAN>
</A></P>

<P>
<A HREF="/scripting/default.asp" TARGET="_top">
    <SPAN id="NavText">Scripting</SPAN>
</A></P>

<P>
<A HREF="/db/searchform.html" TARGET="_top">
    <SPAN id="NavText">Tools Database</SPAN>
</A></P>

<P>
<A HREF="/archives/default.asp" TARGET="_top">
    <SPAN id="NavText">Archives</SPAN>
</A></P>

<P>
<A HREF="/bio.shtml"><SPAN id="NavText">About Us</SPAN>
</A></P>

<P>
<A HREF="http://www.beyondhtml.com/" TARGET="_top">
    <SPAN id="NavText">Home</SPAN>
</A></P></TD>

<TD WIDTH="35" HEIGHT="211" BGCOLOR="#FFFFFF" VALIGN="TOP"></TD>
<TD WIDTH="442" HEIGHT="211" BGCOLOR="#FFFFFF" VALIGN="TOP">

<MAP NAME="logosmall">
  <AREA SHAPE="RECT" COORDS="0,1,199,52"
        HREF="http://www.beyondhtml.com"
        ALT="Jump to Home page"></AREA>
  <AREA SHAPE="default" HREF="http://www.beyondhtml.com"></AREA>
```

```
    </MAP>

    <P>
      <BR></BR> <IMG SRC="/images/logo.gif"
                  ALT="BeyondHTML Logo" ALIGN="MIDDLE"
                  WIDTH="426" HEIGHT="112">
            </IMG>
    </P>
<!-- template rules go here -->

                <Span ID="BoxCopy">
                    <xsl:value-of select="article/headline"/>
                </Span><BR></BR>

                <DIV Class="deck">
                    <xsl:value-of select="article/deck"/>
                </DIV>
                <BR></BR>

                <DIV Class="byline">
                    <xsl:value-of select="article/byline"/>
                </DIV>
                <BR></BR>
                <DIV Class="aBody">
                    <P><xsl:value-of select="article//aBody/para1"/></P>
                    <P><xsl:value-of select="article//aBody/para"/></P>
                    <P><xsl:value-of select="article//aBody/para2"/></P>
                </DIV>
          </TD></TR>
      </TABLE>

      <P ALIGN="CENTER"><FONT SIZE="-1"> [<A HREF="/present/index.xml">Presentation</A>]
      [<A HREF="/markup/index.xml">Markup</A>] [<A HREF="/markup/xml/index.xml">XML</A>]
      <A HREF="/scripting/index.xml">[Scripting</A>] [<A HREF="/db/searchform.html">Tools
        Database</A>] [<A HREF="/archives/index.xml">Archives</A>] [<A HREF="/bio.shtml">About
        Us</A>] [<A HREF="http://www.beyondhtml.com/">Home</A>]</FONT> </P>
    <HR></HR>

    <P><FONT SIZE="-1">Copyright (c) 1998, 1999.
      <A HREF="mailto:mfloyd@beyondhtml.com">Michael Floyd</A> . All Rights Reserved</FONT>
    </P>

    <P><FONT SIZE="-1"><I>Last Modified:

      <!--#echo var="LAST_MODIFIED"--></I></FONT></P>

      </BODY>
    </HTML>
  </xsl:template>
```

```
<xsl:template match="aBody">
    <P>
        <xsl:apply-templates/>
    </P>
</xsl:template>

<xsl:template match="dropCap">
    <DIV Class="dropCap">
        <xsl:apply-templates />
    </DIV>
</xsl:template>

<xsl:template match="bold">
    <B>
        <xsl:apply-templates/>
    </B>
</xsl:template>

<xsl:template match="italic">
    <I>
        <xsl:apply-templates/>
    </I>
</xsl:template>

<xsl:template match="byline[@Email]">
    <A HREF="mailto:mfloyd@BeyondHTML.com">
        <xsl:apply-templates/>
     </A>
</xsl:template>

 <!-- template rules end here -->

</xsl:stylesheet>
```

## 20.3 小　　结

在学习了先前各章的内容之后，进行一个网站的开发实践是一件非常有意义并且令人感兴趣的工作。我们将 XML 作为一种解决方案，应用于网站的实现，向读者展示了如何利用这一先进技术来进行实际的 Web 站点的编写和维护。

XML 是 Web 的未来，也许两三年后，所有的 Web 站点都基于 XML 进行开发，各种各样的信息分发需求都能够得到最大的满足，海量的网络资源可以用标准的规范进行交换和共享，也许……。

XML 的世界是神奇而多变的，探索的过程是有趣且具挑战性的，让我们一起进入 XML 的未来之旅吧！