

Pushing Web Sites with CDF

This chapter covers Microsoft's Channel Definition Format (CDF), which is an XML application for defining channels. A channel is a set of Web pages that can be pushed to a subscriber automatically. A CDF document lists the pages to be pushed, the means by and the frequency with which they're pushed, and similar information. Readers can subscribe to channels using Internet Explorer 4.0 and later. As well as Web pages, channels can use Dynamic HTML, Java, and JavaScript to create interactive, continually updated stock tickers, sports score boxes, and the like. Subject to security restrictions, channels can even push software updates to registered users and install them automatically.

What Is CDF?

The Channel Definition Format (CDF) is an XML application developed at Microsoft for defining channels. Channels enable Web sites to automatically notify readers of changes to critical information. This method is sometimes called *Webcasting* or *push*. Currently, Internet Explorer is the only major browser that implements CDF and broader adoption seems unlikely. The W3C has not done more than formally acknowledge receipt of the CDF specification, and they seem unlikely to do more in the future.

A CDF file is an XML document, separate from, but linked to, the HTML documents on a site. The CDF document defines the parameters for a connection between the readers and the content on the site. The data can be transferred through *push*—sending notifications, or even entire Web sites to registered readers—or through *pull*—readers choose to load the page in their Web browser and get the update information.



In This Chapter

What is CDF?

How channels are created

Description of the channel

Information update schedules

Techniques for precaching and Web crawling

Reader access log

The BASE attribute

The LASTMOD attribute

The USAGE element



You do not need to rewrite your site to take advantage of CDF. The CDF file is simply an addition to the site. A link to a CDF file, generally found on a site's home page downloads a copy of the channel index to the reader's machine. This places an icon on the reader's channel bar, which can be clicked to access the current contents of the channel.

How Channels Are Created

To establish a channel, follow these three steps:

1. Decide what content to include in the channel.
2. Write the channel definition file that identifies this content.
3. Link from the home page of the Web site to the channel-definition file.

Determining Channel Content

Before you get bogged down in the nitty-gritty technical details of creating a channel with CDF, you first have to decide what content belongs in the channel and how it should be delivered.

Your first consideration when converting existing sites to channels is how many and which pages to include. Human interface factors suggest that no channel should have more than eight items for readers to choose from. Otherwise, readers will become confused and have trouble finding what they need. However, channels can be arranged hierarchically. Additional levels of content can be added as sub-channels. For example, a newspaper channel might have sections for business, science, entertainment, international news, national news, and local news. The entertainment section might be divided into sub-channels for television, movies, books, music, and art.

The organization and hierarchy you choose may or may not match the organization and hierarchy of your existing Web site, just as the organization and hierarchy of your Web site does not necessarily match the organization and hierarchy of the files on the server hard drive. However, matching the hierarchy of the channel to the hierarchy of the Web site will make the channel easier to maintain. Nonetheless, you can certainly select particular pages out of the site and arrange them in a hierarchy specific to the channel if it seems logical.

Your second consideration is the way new content will be delivered to subscribers. When subscribing to a channel, readers are offered a choice from three options:

1. The channel can be added to the channel bar and subscribers can check in when they feel like it.

2. Subscribers can be notified of new content via email and then load the channel when they feel like it.
3. The browser can periodically check the site for updates and download the changed content automatically.

Your content should be designed to work well with whichever of these three options the reader chooses.

Creating CDF Files and Documents

Once you've decided what content will be in your channel, and how that content will be organized and delivered, you're ready to write the CDF document that implements these decisions. A CDF document contains identifying information about the contents, schedule, and logos for the channel. All of this information is marked up using a particular set of XML tags. The resulting document is a well-formed XML file. This document will be placed on the Web server where clients can download it.

**Note**

While it would be almost trivial to design a DTD for CDF, and while I suspect Microsoft has one internally, they have not yet published it for the current version of CDF. A DTD for a much earlier and obsolete version of CDF can be found in a W3C note at <http://www.w3.org/TR/NOTE-CDFsubmit.html>. However, this really doesn't come close to describing the current version of CDF. Consequently, CDF documents can be at most well-formed, but not valid.

A CDF document begins with an XML declaration because a CDF document is an XML document and follows the same rules as all XML documents. The root and only required element of a CDF document is `CHANNEL`. The `CHANNEL` element must have an `href` attribute that specifies the page being monitored for changes. The root `CHANNEL` element usually identifies the key page in the channel. Listing 21-1 is a simple CDF document that points to a page that is updated more or less daily.

Listing 21-1: The simplest possible CDF document for a page

```
<?xml version="1.0"?>
<CHANNEL href="http://metalab.unc.edu/xml/index.html">
</CHANNEL>
```

**Note**

Most Microsoft documentation for CDF is based on a pre-release of the XML specification that used the uppercase `<?XML version="1.0"?>` instead of the now current lowercase `<?xml version="1.0"?>`. However, both case conventions seem to work with Internet Explorer, so in this chapter I'll use the lowercase `xml` that conforms to standard XML usage.

As well as the main page, most channels contain a collection of other pages identified by ITEM children. Each ITEM has an HREF attribute pointing to the page. Listing 21-2 demonstrates a channel that contains a main page (<http://metalab.unc.edu/xml/index.html>) with three individual sub-pages in ITEM elements. Channels are often shown in a collapsible outline view that allows the user to show or hide the individual items in the channel as they choose. Figure 21-1 shows this channel expanded in Internet Explorer 5.0's Favorites bar.

Listing 21-2: A CDF channel with ITEM children

```
<?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/maillinglists.html">
  </ITEM>
</CHANNEL>
```

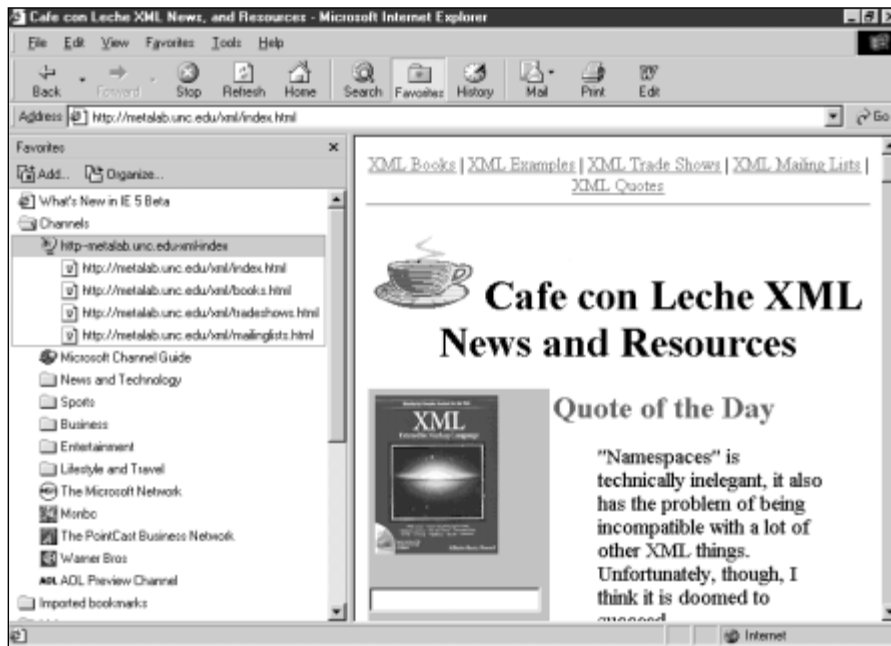


Figure 21-1: The open Channels folder in Internet Explorer 5.0's favorites bar with three sub-pages displayed

The third and final step is to make the CDF file available to the reader. To do this, you provide a link from the Web page to the CDF file. The simplest way to accomplish this is with a standard HTML `A` element that readers click to activate. Generally, the contents of this element will be some text or an image asking the reader to subscribe to the channel. For example:

```
<A HREF="cafeconleche.cdf">Subscribe to Cafe con Leche</A>
```

When the reader activates this link in a CDF-enabled browser (which is just a fancy way of saying Internet Explorer 4.0 and later), the browser downloads the CDF file named in the `HREF` attribute and adds the channel to its list of subscriptions. Other browsers that don't support CDF will probably ask the user to save the document as shown in Figure 21-2.

Once the CDF file has been downloaded, the browser will ask the user how they wish to be notified of future changes to the channel as shown in Figure 21-3. The user has three choices:

1. The channel can be added to the browser and active desktop channel bars. The subscriber must manually select the channel to get the update. This isn't all that different from a bookmark, except that when the user opens the "channel mark," all pages in the channel are refreshed rather than just one.
2. The browser periodically checks the channel for updates and notifies the subscriber of any changes via email. The user must still choose to download the new content.
3. The browser periodically checks the channel for updates and notifies the subscriber of any changes via email. However, when a change is detected, the browser automatically downloads and caches the new content so it's immediately available for the user to view, even if they aren't connected to the Internet when they check the channel site.

Listing 21-2 only makes the first choice available because this particular channel doesn't provide a schedule for update, but we'll add that soon.

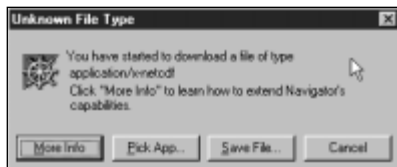


Figure 21-2: Netscape Navigator 5.0 does not support CDF nor understand CDF files.

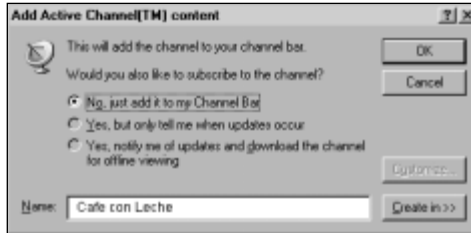


Figure 21-3: Internet Explorer 4.0 asks the user to choose how they wish to be notified of changes at the site.

Description of the Channel

The channel itself and each item in the channel can have a title, an abstract, and up to three logos of different sizes. These are established by giving the `CHANNEL` and `ITEM` elements `TITLE`, `ABSTRACT`, and `LOGO` children.

Title

The title of the channel is not the same as the title of the Web page. Rather, the channel title appears in the channel guide, the channel list, and the channel bar, as shown in Figure 21-1 where the title is `http—metalab.unc.edu-xml-index` (though the subscriber did have the option to customize it by typing a different title as shown in Figure 21-3). You can provide a more descriptive default title for each `CHANNEL` and `ITEM` element by giving it a `TITLE` child. Each `TITLE` element can contain only character data, no markup. Listing 21-3 adds titles to the individual pages in the Cafe con Leche channel as well as to the channel itself. Figure 21-4 shows how this affects the individual items in the channel list.

Listing 21-3: A CDF channel with titles

```
<?xml version="1.0"?>
<CHANNEL HREF="http://metalab.unc.edu/xml/index.html">
  <TITLE>Cafe con Leche</TITLE>
  <ITEM HREF="http://metalab.unc.edu/xml/books.html">
    <TITLE>Books about XML</TITLE>
  </ITEM>
  <ITEM HREF="http://metalab.unc.edu/xml/tradeshows.html">
    <TITLE>Trade shows and conferences about XML</TITLE>
  </ITEM>
  <ITEM HREF="http://metalab.unc.edu/xml/maillinglists.html">
    <TITLE>Mailing Lists dedicated to XML</TITLE>
  </ITEM>
</CHANNEL>
```

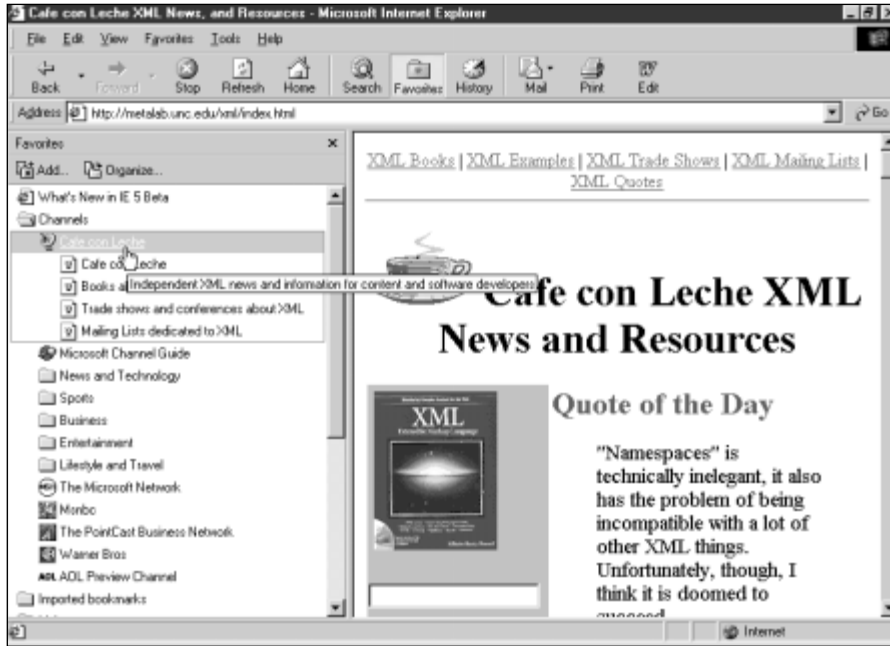


Figure 21-4: Titles are shown in the channels bar and abstracts are shown in tool tips.

Abstract

Titles may be sufficient for a channel with a well-established brand like Disney or MSNBC, but for the rest of us lesser lights in the news firmament it probably doesn't hurt to tell subscribers a little more about what they can expect to find at a given site. To this end, each CHANNEL and ITEM element can contain a single ABSTRACT child element. The ABSTRACT element should contain a short (200 characters or less) block of text describing the item or channel. Generally, this description will appear in a tool-tip window as shown in Figure 21-4, which is based on Listing 21-4.

Listing 21-4: A CDF channel with titles and abstracts

```
<?xml version="1.0"?>
<CHANNEL HREF="http://metalab.unc.edu/xml/index.html">
  <TITLE>Cafe con Leche</TITLE>
  <ABSTRACT>
    Independent XML news and information for content
    and software developers
  </ABSTRACT>
```

Continued

Listing 21-4 (continued)

```
<ITEM HREF="http://metalab.unc.edu/xml/books.html">
  <TITLE>Books about XML</TITLE>
  <ABSTRACT>
    A comprehensive list of books about XML
    with capsule reviews and ratings
  </ABSTRACT>
</ITEM>

<ITEM HREF="http://metalab.unc.edu/xml/tradeshows.html">
  <TITLE>Trade shows and conferences about XML</TITLE>
  <ABSTRACT>
    Upcoming conferences and shows with an XML focus
  </ABSTRACT>
</ITEM>

<ITEM HREF="http://metalab.unc.edu/xml/maillinglists.html">
  <TITLE>Mailing Lists dedicated to XML</TITLE>
  <ABSTRACT>
    Mailing lists where you can discuss XML
  </ABSTRACT>
</ITEM>
</CHANNEL>
```

Logos

CDF documents can specify logos for channels. These logos appear on the reader's machine; either on the desktop or in the browser's channel list. Logos can be used in a number of different ways within the channel: icons on the desktop, icons in the program launcher, and logos in the channel guide and channel bar. Each CHANNEL and ITEM element can have up to three logos: one for the desktop, one for the program launcher, and one for the channel bar.

A particular logo is attached to a channel with the LOGO element. This element is a child of the CHANNEL it represents. The HREF attribute of the LOGO element is an absolute or relative URL where the graphic file containing the logo is found. Internet Explorer supports GIF, JPEG, and ICO format images for logos — but not animated GIFs. Because logos may appear against a whole range of colors and patterns on the desktop, GIFs with a transparent background that limit themselves to the Windows halftone palette work best.

The `LOGO` element also has a required `STYLE` attribute that specifies the size of the image. The value of the `STYLE` attribute must be one of the three keywords `ICON`, `IMAGE`, or `IMAGE-WIDE`. These are different sizes of images, as given in Table 21-1. Figure 21-5 shows the logos used for Cafe con Leche in the three different sizes.

Table 21-1
Values for the `STYLE` Attribute of the `LOGO` Element

<i>Image Size Value</i>	<i>Description</i>
<code>ICON</code>	A 16-pixel-wide by 16-pixel-high icon displayed in the file list and in the channel bar next to the child elements in a hierarchy, as shown in Figure 21-2.
<code>IMAGE</code>	An 80-pixel-wide by 32-pixel-high image displayed in the desktop channel bar.
<code>IMAGE-WIDE</code>	A 194-pixel-wide by 32-pixel-high image displayed in the browser's channel bar. If a hierarchy of channels is nested underneath, they appear when the reader clicks this logo, as shown in Figure 21-3.



Figure 21-5: The Cafe con Leche channel icons in three different sizes

When the content in the channel changes, the browser places a highlight gleam in the upper-left corner of the logo image. This gleam hides anything in that corner. Also, if a reader stretches the window width beyond the recommended 194 pixels, the browser uses the top-right pixel to fill the expanded logo. Consequently you need to pay special attention to the upper-left and right corners of the logo.

Information Update Schedules

The `CHANNEL`, `TITLE`, `ABSTRACT`, and `LOGO` elements are enough to build a working channel, but all they provide is a visible connection that readers can use to pilot themselves quickly to your site. However, you don't have any means to push content to the readers. Passive channels — that is, channels like Listings 21-1 through 21-5 that don't have an explicit push schedule — don't do very much.

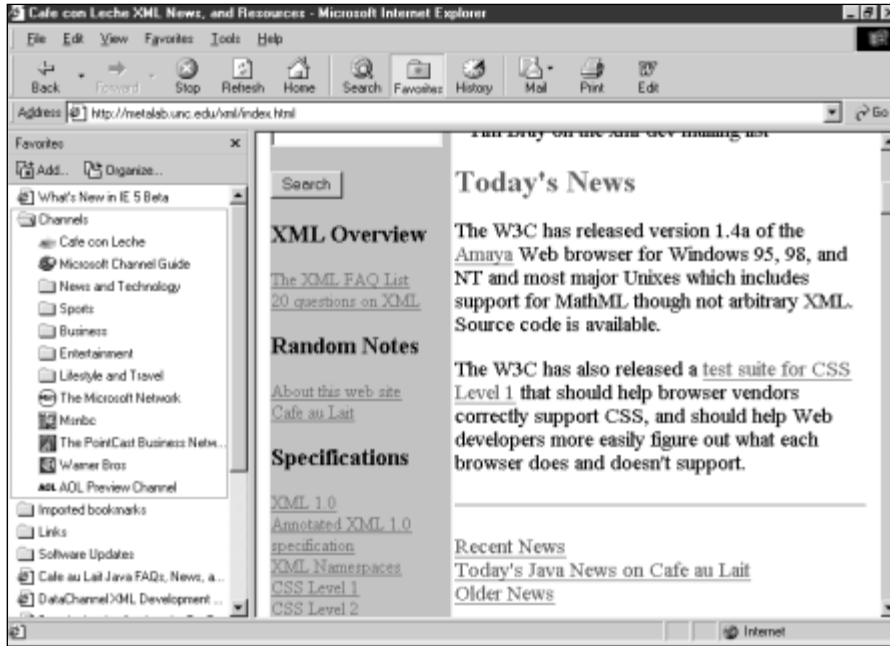


Figure 21-6: The favorites bar now contains the Cafe con Leche icons instead of the generic channel icon.

Listing 21-5 is a CDF document that provides various sizes of logos. Figure 21-6 shows the Internet Explorer 5.0 favorites bar with the new Cafe con Leche logo.

Listing 21-5: A CDF channel with various sizes of logos

```
<?xml version="1.0"?>
<CHANNEL HREF="http://metalab.unc.edu/xml/index.html">
  <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="http://metalab.unc.edu/xml/books.html">
    <TITLE>Books about XML</TITLE>
    <ABSTRACT>
      A comprehensive list of books about XML
      with capsule reviews and ratings
    </ABSTRACT>
  </ITEM>
</CHANNEL>
```

```

<ITEM HREF="http://metalab.unc.edu/xml/tradeshows.html">
  <TITLE>Trade shows and conferences about XML</TITLE>
  <ABSTRACT>
    Upcoming conferences and shows with an XML focus
  </ABSTRACT>
</ITEM>

<ITEM HREF="http://metalab.unc.edu/xml/maillinglists.html">
  <TITLE>Mailing Lists dedicated to XML</TITLE>
  <ABSTRACT>
    Mailing lists where you can discuss XML
  </ABSTRACT>
</ITEM>
</CHANNEL>

```

To actually push the contents to subscribers, you have to include scheduling information for updates. You can schedule the entire channel as one or schedule individual items in the channel separately. This is accomplished by adding a SCHEDULE child element to the channel. For example:

```

<SCHEDULE STARTDATE="1998-03-29" STOPDATE="1999-03-29"
  TIMEZONE="-0500">
  <INTERVALTIME DAY="7"/>
  <EARLIESTTIME DAY="1" HOUR="0" MIN="0"/>
  <LATESTTIME DAY="2" HOUR="12" MIN="0"/>
</SCHEDULE>

```

The SCHEDULE element has three attributes: STARTDATE, STOPDATE, and TIMEZONE. The STARTDATE indicates when the schedule begins and STOPDATE indicates when it ends. Target the period between your usual site overhauls. If you change the structure of your Web site on a regular interval, use that interval. STARTDATE and STOPDATE use the same date format: full numeric year, two-digit numeric month, and two-digit day of month; for example 1999-12-31.

The TIMEZONE attribute shows the difference in hours between the server's time zone and Greenwich Mean Time. If the tag does not include the TIMEZONE attribute, the scheduled update occurs according to the reader's time zone — not the server's. In the continental U.S., Eastern Standard Time is -0500, Central Standard Time is -0600, Mountain Standard Time is -0700, and Pacific Standard Time is -0800. Hawaii and Alaska are -1000.

SCHEDULE can have between one and three child elements. INTERVALTIME is a required, empty element that specifies how often the browser should check the channel for updates (assuming the user has asked the browser to do so). INTERVALTIME has DAY, HOUR, and MIN attributes. The DAY, HOUR, and MIN attributes are added to calculate the amount of time that is allowed to elapse between updates. As long as one is present, the other two can be omitted.

`EARLIESTTIME` and `LATESTTIME` are optional elements that specify times between which the browser should check for updates. The updates and resulting server load are distributed over the interval between the earliest and latest times. If you don't specify these, the browser simply checks in at its convenience. `EARLIESTTIME` and `LATESTTIME` have `DAY` and `HOURL` attributes used to specify when updates take place. `DAY` ranges from 1 (Sunday) to 7 (Saturday). `HOURL` ranges from 0 (midnight) to 23 (11:00 P.M.). For instance, the above example says that the browser should update the channel once a week (`INTERVALTIME DAY="7"`) between Sunday midnight (`EARLIESTTIME DAY="1" HOURL="0"`) and noon Monday (`LATESTTIME DAY="2" HOURL="12"`).

`EARLIESTTIME` and `LATESTTIME` may also have a `TIMEZONE` attribute that specifies the time zone in which the earliest and latest times are calculated. If a time zone isn't specified, the reader's time zone is used to determine the earliest and latest times. To force the update to a particular time zone, include the optional `TIMEZONE` attribute in the `EARLIESTTIME` and `LATESTTIME` tags. For example:

```
<EARLIESTTIME DAY="1" HOURL="0" TIMEZONE="-0500"/>
<LATESTTIME DAY="2" HOURL="12" TIMEZONE="-0500"/>
```

To push an update across a LAN, you can choose the day of the week (for example, Sunday) and the time span (midnight to five in the morning). All browsers update during that five-hour period. If you update across Internet connections, your readers have to be connected to the Internet for the browser to update the channel.

Listing 21-6 expands the Cafe con Leche channel to include scheduled updates. Since content is updated almost daily `INTERVALTIME` is set to one day. Most days the update takes place between 7:00 a.m. and 12:00 noon Eastern time. Consequently, it sets `EARLIESTTIME` to 10:00 a.m. EST and `LATESTTIME` to 12:00 noon EST. There's no particular start or end date for the changes to this content, so the `STARTDATE` and `STOPDATE` attributes are omitted from the schedule.

Listing 21-6: A CDF channel with scheduled updates

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

  <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"/>

  <SCHEDULE TIMEZONE="-0500">
```

```

<INTERVALTIME DAY="1"/>
<EARLIESTTIME HOUR="10" TIMEZONE="-0500"/>
<LATESTTIME HOUR="12" TIMEZONE="-0500"/>
</SCHEDULE>

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

<ITEM HREF="http://metalab.unc.edu/xml/tradeshows.html">
<TITLE>Trade shows and conferences about XML</TITLE>
<ABSTRACT>
  Upcoming conferences and shows with an XML focus
</ABSTRACT>
</ITEM>

<ITEM HREF="http://metalab.unc.edu/xml/maillinglists.html">
<TITLE>Mailing Lists dedicated to XML</TITLE>
<ABSTRACT>
  Mailing lists where you can discuss XML
</ABSTRACT>
</ITEM>
</CHANNEL>

```

Precaching and Web Crawling

If the subscriber has chosen to download the channel's contents automatically when they change, then the site owner has the option of allowing subscribers to view the pages offline and even to download more than merely those pages identified in the CDF document. In particular, you can allow the browser to spider through your site, downloading additional pages between one and three levels deep from the specified pages.

Precaching

By default, browsers precache the pages listed in a channel for offline browsing if the user has requested that they do so. However, the author can prevent a page from being precached by including a `PRECACHE` attribute in the `CHANNEL` or `ITEM` element with the value `NO`. For example:

```

<CHANNEL PRECACHE="NO"
  HREF="http://metalab.unc.edu/xml/index.html">
  ...
</CHANNEL>

```

If the value of `PRECACHE` is `NO`, then the content will not be precached regardless of user settings. If the value of `PRECACHE` is `YES` (or there is no explicit `PRECACHE` attribute) *and* the user requested precaching when they subscribed, then the content will be downloaded automatically. However, if the user has not requested precaching, then the site channel will not be precached regardless of the value of the `PRECACHE` attribute.

When you design a channel, you must keep in mind that some readers will view content offline almost exclusively. As a result, any links in the channel contents are effectively dead. If you are pushing documents across an intranet, the cache option doesn't make a lot of sense, as you'll be duplicating the same files on disks across the corporation. If you are delivering content to readers who pay for online time, you may want to organize it so that it can be cached and easily browsed offline.

Web Crawling

Browsers are not limited to loading only the Web pages specified in `CHANNEL` and `ITEM` elements. If a `CHANNEL` or `ITEM` element has a `LEVEL` attribute with a value higher than zero, the browser will Web crawl during updates. Web crawling lets the browser collect more pages than are listed in the channel. For example, if the page listed in a channel contains a number of links to related topics, it may be easier to let the browser load them all rather than list them in individual `ITEM` elements. If the site has a fairly even hierarchy, you can safely add a `LEVEL` attribute to the top-most channel tag and allow the Web crawl to include all of the pages at the subsequent levels. `LEVEL` can range from zero (the default) to three, which indicates how far down into the link hierarchy you want the browser to dig when caching the content. The hierarchy is the abstract hierarchy defined by the document links, not the hierarchy defined by the directory structure of files on the Web server. Framed pages are considered to be at the same level as the frameset page, even though an additional link is required for the former. The `LEVEL` attribute really only has meaning if precaching is enabled.

Listing 21-7 sets the `LEVEL` of the Cafe con Leche channel to 3. This goes deep enough to reach every page on the site. Since the pages previously referenced in `ITEM` children are only one level down from the main page, there's not as much need to list them separately. However, Web-crawling this deep may not be such a good idea. Most of the pages on the site don't change daily. Nonetheless, they'll still be checked each and every update.

Listing 21-7: A CDF channel that precaches three levels deep

```
<?xml version="1.0"?>
<CHANNEL LEVEL="3"
  HREF="http://metalab.unc.edu/xml/index.html">
```

```

<TITLE>Cafe con Leche</TITLE>
<ABSTRACT>
  Independent XML news and information for content
  and software developers
</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"/>

<SCHEDULE TIMEZONE="-0500">
  <INTERVALTIME DAY="1"/>
  <EARLIESTTIME HOUR="10" TIMEZONE="-0500"/>
  <LATESTTIME HOUR="12" TIMEZONE="-0500"/>
</SCHEDULE>

</CHANNEL>

```

Reader Access Log

One disadvantage to channels compared to traditional Web surfing is that the server does not necessarily know which pages the surfer actually saw. This can be important for tracking advertisements, among other things. Internet Explorer can track the reader's passage through a site cached offline, and report it back to the Web server. However, the user always has the option to disable this behavior if they feel it's a privacy violation.

To collect statistics about the offline browsing of a site, you add `LOG` and `LOGTARGET` child elements to the `CHANNEL` element. During a channel update, the server sends the new channel contents to the browser while the browser sends the log file to the server. The `LOG` element always has this form, though other possible values of the `VALUE` attribute may be added in the future:

```
<LOG VALUE="document:view"/>
```

The `LOGTARGET` element has an `HREF` attribute that identifies the URL it will be sent to, a `METHOD` attribute that identifies the HTTP method like `POST` or `PUT` that will be used to upload the log file, and a `SCOPE` attribute that has one of the three values: `ALL`, `ONLINE`, or `OFFLINE` indicating which page views should be counted. The `LOGTARGET` element may have a `PURGETIME` child with an `HOUR` attribute that specifies the number of hours for which the logging information is considered valid. It may also have any number of `HTTP-EQUIV` children used to set particular key-value pairs in the HTTP MIME header. Listing 21-8 demonstrates a channel with a reader-access log.

Listing 21-8: A CDF channel with log reporting

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

  <TITLE>Cafe con Leche</TITLE>
  <ABSTRACT>
    Independent XML news and information for content
    and software developers
  </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"/>

  <LOG VALUE="document:view"/>
  <LOGTARGET METHOD="POST" SCOPE="ALL"
    HREF="http://metalab.unc.edu/xml/cgi-bin/getstats.pl" >
    <PURGETIME HOUR="12"/>
    <HTTP-EQUIV NAME="ENCODING-TYPE" VALUE="text"/>
  </LOGTARGET>

  <SCHEDULE TIMEZONE="-0500">
    <INTERVALTIME DAY="1"/>
    <EARLIESTTIME HOUR="10" TIMEZONE="-0500"/>
    <LATESTTIME HOUR="12" TIMEZONE="-0500"/>
  </SCHEDULE>

  <ITEM HREF="http://metalab.unc.edu/xml/books.html">
    <TITLE>Books about XML</TITLE>
    <ABSTRACT>
      A comprehensive list of books about XML
      with capsule reviews and ratings
    </ABSTRACT>
    <LOG VALUE="document:view"/>
  </ITEM>

  <ITEM HREF="http://metalab.unc.edu/xml/tradeshows.html">
    <TITLE>Trade shows and conferences about XML</TITLE>
    <ABSTRACT>
      Upcoming conferences and shows with an XML focus
    </ABSTRACT>
    <LOG VALUE="document:view"/>
  </ITEM>

  <ITEM HREF="http://metalab.unc.edu/xml/maillinglists.html">
    <TITLE>Mailing Lists dedicated to XML</TITLE>
    <ABSTRACT>
      Mailing lists where you can discuss XML
    </ABSTRACT>
  </ITEM>
</CHANNEL>
```

Only elements with LOG children will be noted in the log file. For instance, in Listing 21-8, hits to `http://metalab.unc.edu/xml/index.html`, `http://metalab.unc.edu/xml/books.html`, and `http://metalab.unc.edu/xml/tradeshows.html` will be logged. However hits to `http://metalab.unc.edu/xml/maillinglists.html` will not be.

The CDF logging information is stored in the Extended File Log format used by most modern Web servers. However, the Web server must be configured, most commonly through a CGI program, to accept the log file the client sends and merge it into the main server log.

The LOGTARGET element should appear as a child of the top-level CHANNEL tag, and describes log file handling for all items it contains. However, each CHANNEL and ITEM element that you want included in the log must include its own LOG child.

The BASE Attribute

The previous examples have all used absolute URLs for CHANNEL and ITEM elements. However, absolute URLs are inconvenient. For one thing, they're often long and easy to mistype. For another, they make site maintenance difficult when pages are moved from one directory to another, or from one site to another. You can use relative URLs instead if you include a BASE attribute in the CHANNEL element.

The value of the BASE attribute is a URL to which relative URLs in the channel can be relative. For instance, if the BASE is set to "`http://metalab.unc.edu/xml/`", then an HREF attribute can simply be "`books.html`" instead of "`http://metalab.unc.edu/xml/books.html`". Listing 21-9 demonstrates.

Listing 21-9: A CDF channel with a BASE attribute

```
<?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
```

Continued

Listing 21-9 (continued)

```

        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>

```

Whichever location you use for the link to the content, you can use a relative URL in the child elements if you specify a BASE attribute in the parent CHANNEL element. The BASE attribute also changes the hierarchy display in Internet Explorer. The base page will display in the browser window when child elements are not associated with a page.

The LASTMOD Attribute

When a browser makes a request of a Web server, the server sends a MIME header along with the requested file. This header includes various pieces of information like the MIME type of the file, the length of the file, the current date and time, and the time the file was last modified. For example:

```

HTTP/1.1 200 OK
Date: Wed, 27 Jun 1999 21:42:31 GMT
Server: Stronghold/2.4.1 Apache/1.3.3 C2NetEU/2409 (Unix)
Last-Modified: Tue, 20 Oct 1998 13:15:36 GMT
ETag: "4b94d-c70-362c8cf8"
Accept-Ranges: bytes
Content-Length: 3184
Connection: close
Content-Type: text/html

```

If a browser sends a HEAD request instead of the more common GET request, only the header is returned. The browser can then inspect the Last-Modified header to determine whether a previously loaded file from the channel needs to be reloaded or not. However, although HEAD requests are quicker than GET requests, a lot of them still eat up server resources.

To cut down on the load that frequent channel updates place on your server, you can add LASTMOD attributes to all CHANNEL and ITEM tags. The browser will only have to check back with the server for modification times for those items and channels that don't provide LASTMOD attributes.

The value of the LASTMOD attribute is a date and time in a *year-month-dayThour:minutes* form like 2000-05-23T21:42-when the page referenced by the HREF attribute was last changed. The browser detects and compares the LASTMOD date given in the CDF file with the last modified date provided by the Web server. When the content on the Web server has changed, the cache is updated with the current content. This way the browser only needs to check one file, the CDF document, for modification times rather than every file that's part of the channel. Listing 21-10 demonstrates.

Listing 21-10: A CDF channel with LASTMOD attributes

```
<?xml version="1.0"?>
<CHANNEL BASE="http://metalab.unc.edu/xml/"
          LASTMOD="1999-01-27T12:16" >
  <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" LASTMOD="1999-01-03T16:25">
    <TITLE>Books about XML</TITLE>
    <ABSTRACT>
      A comprehensive list of books about XML
      with capsule reviews and ratings
    </ABSTRACT>
  </ITEM>

  <ITEM HREF="tradeshows.html" LASTMOD="1999-01-10T11:40">
    <TITLE>Trade shows and conferences about XML</TITLE>
    <ABSTRACT>
      Upcoming conferences and shows with an XML focus
```

Continued

Listing 21-10 (continued)

```
</ABSTRACT>
</ITEM>

<ITEM HREF="mailinglists.html" LASTMOD="1999-01-06T10:50">
  <TITLE>Mailing Lists dedicated to XML</TITLE>
  <ABSTRACT>
    Mailing lists where you can discuss XML
  </ABSTRACT>
</ITEM>
</CHANNEL>
```

In practice, this is way too much trouble to do manually, especially for frequently changed documents (and the whole point of channels and push is that they provide information that changes frequently). However, you might be able to write the CDF document as a file full of server-side includes that automatically incorporate `LASTMOD` values in the appropriate format or devise some other programmatic solution rather than manually adjusting the `LASTMOD` attribute every time you edit a file.

The USAGE Element

A `CHANNEL` or `ITEM` element may contain an optional `USAGE` child element that extends the presence of the channel on the subscriber's desktop. The meaning of the `USAGE` element is determined by its `VALUE` attribute. Possible values for the `VALUE` attribute are:

- ♦ Channel
- ♦ DesktopComponent
- ♦ Email
- ♦ NONE
- ♦ ScreenSaver
- ♦ SoftwareUpdate

Most of the time `USAGE` is an empty element. For example:

```
<USAGE VALUE="ScreenSaver" />
```

The default value for `USAGE` is `Channel`. Items with channel usage appear in the browser channel bar. All the `CHANNEL` and `ITEM` elements you've seen until now have had `Channel` usage, even though they didn't have an explicit usage element. Other values for `USAGE` allow different user interfaces to the channel content.

DesktopComponent Value

Desktop components are small Web pages or images that are displayed directly on the user's desktop. Since a Web page can contain a Java applet, fancy DHTML, or an ActiveX control, a desktop component can actually be a program (assuming the subscriber has abandoned all semblance of caution and installed Active Desktop).

The desktop component is installed on the subscriber's desktop with a separate CDF document containing an `ITEM` element that points to the document to be displayed on the user's desktop. As well as the usual child elements, this `ITEM` must contain a non-empty `USAGE` element whose `VALUE` is `DesktopComponent`. This `USAGE` element may contain `OPENAS`, `HEIGHT`, `WIDTH`, and `CANRESIZE` children.

The `VALUE` attribute of the `OPENAS` element specifies the type of file at the location in the `ITEM` element's `href` attribute. This should either be `HTML` or `Image`. If no `OPENAS` element is present, Internet Explorer assumes it is an `HTML` file.

The `VALUE` attributes of the `HEIGHT` and `WIDTH` elements specify the number of pixels the item occupies on the desktop.

The `VALUE` attribute of the `CANRESIZE` element indicates whether the reader can change the height and width of the component on the fly. Its possible values are `Yes` and `No`. `Yes` is the default. You can also allow or disallow horizontal or vertical resizing independently with `CANRESIZEX` and `CANRESIZEY` elements.

Listing 21-11 is a simple desktop component that displays a real time image of the Sun as provided by the friendly folks at the National Solar Observatory in Sunspot, New Mexico (<http://vtt.sunspot.noao.edu/gifs/video/sunnow.jpg>). The image is 640 pixels high, 480 pixels wide, but resizable. The image is refreshed every minute between 6:00 a.m. MST and 7:00 p.m. MST. (There's no point refreshing the image at night.)

Listing 21-11: A DesktopComponent channel

```
<?xml version="1.0"?>
<CHANNEL href="http://vtt.sunspot.noao.edu/sunpic.html">
  <TITLE>
    Hydrogen Alpha Image of the Sun Desktop Component
  </TITLE>
  <ABSTRACT>
```

Continued

Listing 21-11 (continued)

This desktop component shows a picture of the Sun as it appears this very minute from the top of Sacramento Peak in New Mexico. The picture is taken in a single color at the wavelength of the Hydrogen alpha light (6563 Angstroms) using a monochrome camera which produces a greyscale image in which the red light of Hydrogen alpha appears white.

```
</ABSTRACT>

<ITEM
  HREF="http://vtt.sunspot.noao.edu/gifs/video/sunnow.jpg">
  <TITLE>Hydrogen Alpha Image of the Sun</TITLE>

  <SCHEDULE TIMEZONE="-0700">
    <INTERVALTIME MIN="1"/>
    <EARLIESTTIME HOUR="6"/>
    <LATESTTIME HOUR="19"/>
  </SCHEDULE>

  <USAGE VALUE="DesktopComponent">
    <WIDTH VALUE="640"/>
    <HEIGHT VALUE="480"/>
    <CANRESIZE VALUE="Yes"/>
    <OPENAS VALUE="Image"/>
  </USAGE>
</ITEM>
</CHANNEL>
```

Email Value

Normally, when a browser notifies a subscriber of a change to channel content by sending them email, it sends along the main page of the channel as the text of the email message. However, you can specify that a different email message be sent by including an ITEM in the channel whose USAGE element has the value email.

Listing 21-12 specifies that the file at <http://metalab.unc.edu/xml/what-snew.html> will be used to notify subscribers of content changes. If the first ITEM were not present, then <http://metalab.unc.edu/xml/> from the CHANNEL HREF attribute would be used instead. This gives you an opportunity to send a briefer message specifying what has changed, rather than sending the entire changed page. Often “What’s new” information is easier for readers to digest than the entire page.

Listing 21-12: A channel that emails a separate notification

```

<?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>

```

NONE Value

Items whose **USAGE** value is **NONE** don't appear anywhere; not in the channel bar, not on the Active Desktop, not on the favorites menu, nowhere. However, such items are precached and are thus more quickly available for applets and HTML pages that refer to them later.

Precaching channel content is useful for including items — such as sound and video clips — that you want to move to the reader's machine for use by channel pages. You can precache a single item or a series of items by defining a channel that includes the set of precached items, as is demonstrated in this example:

```
<ITEM HREF="welcome.wav"> <USAGE="NONE"/> </ITEM>
<ITEM HREF="spacemusic.au"> <USAGE="NONE"/> </ITEM>
```

This example includes two sound files used at the site when the browser downloads the channel contents for offline viewing. These two files won't be displayed in the channel bar, but if a file in the channel bar does use one of these sound files, then it will be immediately available, already loaded when the page is viewed offline. The reader won't have to wait for them to be downloaded from a remote Web site, an important consideration when dealing with relatively large multimedia files.

ScreenSaver Value

Items whose `USAGE` value is `ScreenSaver` point to an HTML page to replace the normal desktop after a user-specified period of inactivity. Generally, a screen saver will be written as a completely separate CDF document from the normal channel, and will require a separate download and install link. For example:

```
<A HREF="ccl_screensaver.cdf">
  Download and install the Cafe con Leche Screen Saver!
</A>
```

Unless the subscriber has already selected the Channel Screen Saver as the system screen saver in the Display control panel as shown in Figure 21-7, the browser will ask the user whether they want to use the Channel Screen Saver or the currently selected screen saver. Assuming they choose the Channel Screen Saver, the next time the screen is saved, the document referenced in the screen saver channel will be loaded and displayed. If the user has subscribed to more than one screen saver channel, the browser will rotate through the subscribed screen saver channels every 30 seconds. The user can change this interval and a few other options (whether screen savers play sounds, for instance) using the screen saver settings in the Display control panel.

Listing 21-13 is a simple screen saver channel. The actual document displayed when the screen is saved is pointed to by the `ITEM` elements `HREF` attribute. This page will generally make heavy use of DHTML, JavaScript, and other tricks to animate the screen. A static screen saver page is a bad idea.

Listing 21-13: A screen saver channel

```
<?xml version="1.0"?>
<CHANNEL BASE="http://metalab.unc.edu/xml/">

  <ITEM HREF="http://metalab.unc.edu/screensaver.html">
    <USAGE VALUE="ScreenSaver"/>
  </ITEM>

</CHANNEL>
```



Figure 21-7: The Screen Saver tab of the Display Properties control panel in Windows NT 4.0

Two things you should keep in mind when designing screen savers:

1. Presumably the user is doing something else when the screen is saved. After all, inactivity activates the screen saver. Therefore, don't go overboard or expect a lot of user attention or interaction with your screen saver.
2. Although almost no modern display really needs its screen saved, screen savers should save the screen nonetheless. Thus most of the screen should be dark most of the time, and no pixel on the screen should ever be continuously on. Most importantly, no pixel should continuously be one non-black color, especially white.

SoftwareUpdate Value

The final possible value of the `USAGE` element is `SoftwareUpdate`. Channels aren't limited to delivering news and Web pages. They can send software updates too. Software update channels can both notify users of updates to software and deliver the product across the Internet. Given a sufficiently trusting (perhaps insufficiently paranoid is more accurate) user, they can even automatically install the software.

To create a software push channel, write a CDF file with a root `CHANNEL` element whose `USAGE` element has the value `SoftwareUpdate`. This channel can have a title, abstract, logos, and schedule, just like any other channel. Listing 21-14 is a fake software update channel.

Listing 21-14: A software update channel

```
<?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>
```

Besides the `VALUE` of the `USAGE` element, the key to a software update channel is its `SOFTPKG` child element. The `HREF` attribute of the `SOFTPKG` element provides a URL from which the software can be downloaded and installed. The URL should point to a compressed archive of the software in Microsoft's cabinet (CAB) format. This archive must carry a digital signature from a certificate authority. Furthermore, it must also contain an OSD file describing the software update. OSD, the

Open Software Description format, is an XML application for describing software updates invented by Microsoft and Marimba. The OSD file structure and language is described on the Microsoft Web site at <http://www.microsoft.com/standards/osd/>.

Cross-Reference

OSD is covered briefly in Chapter 2, *An Introduction to XML Applications*.

The `SOFTPKG` element must also have a `NAME` attribute that contains up to 260 characters describing the application. For example, “WhizzyWriter 2100 with tint control 2.1EA3”.

The `SOFTPKG` element must also have a `STYLE` attribute with one of two values — `ActiveSetup` or `MSICD` (Microsoft Internet Component Download) which determines how the software is downloaded and installed.

There are several optional attributes to `SOFTPKG` as well. The `SOFTPKG` element may have a `PRECACHE` attribute with either the value `Yes` or `No`. This has the same meaning as other `PRECACHE` attributes; that is, determining whether the package will be downloaded before the user decides whether they want it. The `VERSION` attribute is a comma-separated list of major, minor, custom, and build version numbers such as “6,2,3,3124”. Finally, setting the `AUTOINSTALL` attribute to `Yes` tells the browser to download the software package automatically as soon as the CDF document is loaded. The value `No` instructs the browser to wait for a specific user request and is the default if the `AUTOINSTALL` attribute is not included.

These child elements can go inside the `SOFTPKG` element:

- ♦ `TITLE`
- ♦ `ABSTRACT`
- ♦ `LANGUAGE`
- ♦ `DEPENDENCY`
- ♦ `NATIVECODE`
- ♦ `IMPLEMENTATION`

However these elements are not part of CDF. Rather they’re part of OSD. (Technically `SOFTPKG` is too.) Consequently, I’ll only summarize them here:

- ♦ The `TITLE` element of the `SOFTPKG` uses the same options as the standard CDF `TITLE`.
- ♦ The `ABSTRACT` element describes the software and is essentially the same as the CDF `ABSTRACT` element.
- ♦ The `LANGUAGE` element defines the language supported by this update using a `VALUE` attribute whose value is an ISO 639/RFC 1766 two-letter language code

such as EN for English. If multiple languages are supported, they're separated by semicolons.

- ♦ The `DEPENDENCY` element is empty with a single attribute, `ACTION` which may take on one of two values — `Assert` or `Install`. `Assert` is the default and means that the update will only be installed if the necessary CAB file is already on the local computer. With a value of `Install`, the necessary files will be downloaded from the server.
- ♦ The `NATIVECODE` element holds `CODE` child elements. Each `CODE` child element points to the distribution files for a particular architecture such as Windows 98 on X86 or Windows NT on alpha.
- ♦ The `IMPLEMENTATION` element describes the configuration required for the software package. If the requirements described in the implementation tag are not found on the reader's machine, the download and installation do not proceed. The `IMPLEMENTATION` element is an optional element with child elements `CODEBASE`, `LANGUAGE`, `OS`, and `PROCESSOR`.

The `CODEBASE` element has `FILENAME` and `HREF` attributes that say where the files for the update can be found.

The `LANGUAGE` element is the same as the `LANGUAGE` element above.

The `OS` element has a `VALUE` attribute whose value is `Mac`, `Win95`, or `Winnt`, thereby identifying the operating system required for the software. This element can have an empty child element called `OSVERSION` with a `VALUE` attribute that identifies the required release.

The `PROCESSOR` element is an empty element whose `VALUE` attribute can have the value `Alpha`, `MIPS`, `PPC`, or `x86`. This describes the CPU architecture the software supports.

For more details about OSD, you can see the reference at <http://www.microsoft.com/workshop/delivery/osd/reference/reference.asp>, or the specification at <http://www.microsoft.com/standards/osd/default.asp>.

Summary

In this chapter, you learned:

- ♦ The Channel Definition Format (CDF) is an XML application used to describe data pushed from Web sites to Web browsers.
- ♦ CDF files are XML files, although they customarily have the file name extension `.cdf` instead of `.xml`. The root element of a CDF file is `CHANNEL`.
- ♦ Each `CHANNEL` element must contain an `HREF` attribute identifying the pushed page.

- ♦ A CHANNEL element may contain additional ITEM child elements whose HREF attributes contain URLs of additional pages to be pushed.
- ♦ Each CHANNEL and ITEM element may contain TITLE, ABSTRACT, and LOGO children that describe the content of the page the element references.
- ♦ The SCHEDULE element specifies when and how often the browser should check the server for updates.
- ♦ The LOG element identifies items whose viewing is reported back to the Web server, though the subscriber can disable this reporting.
- ♦ The LOGTARGET element defines how logging information from a channel is reported back to the server.
- ♦ The BASE attribute provides a starting point from which relative URLs in child element HREF attributes can be calculated.
- ♦ The LASTMOD attribute specifies the last time a page was changed so the browser can tell whether or not it needs to be downloaded.
- ♦ The USAGE attribute allows you to use Web pages as channels, precached content, Active Desktop components, screen savers, and software updates.

The next chapter explores a completely different application of XML to vector graphics — the Vector Markup Language (VML for short).



