1999-04-13 00:00:00

XML, Part 2: The M.O. on DTD Files

What's a DTD and why do you need one?

Posted by : Elizabeth Powell Crowe

The first rule of computers hasn't changed since the 1960s: Garbage in, garbage out. That's especially true when you work with a programming language--even one as permissive as XML.

In an earlier column, I looked at what you can do with XML. To recap, XML, or extensible markup language, is a subset of SGML that lets you define your own tags and manipulate data without having to learn tricky SGML code. In this column, I'm going to take the first step and examine the document type definition (DTD), which is the foundation of an XML document.

Playing by the Rules

Why are DTDs so important? Because they lay out the rules of the game. Without a DTD, you wouldn't have a template for writing XML Web pages, and XML browsers wouldn't know how to process them. The DTD lets a parser process an XML page, identifying what kinds of information are mandatory, what's optional, where they appear, and how these elements relate to one another. Only then can applications such as browsers, style sheets, and search engines interact with the XML document. In short, the DTD is a guide for both the creator of an XML document and the users who read those pages with XML-capable browsers (such as Internet Explorer 5.0).

The DTD defines the relationships between all the objects within an XML document--chapters, headings, paragraphs, and so forth. XML readers look to the DTD for advance notice of which names and structures will be used in a given document type.

Charles F. Goldfarb, the inventor of SGML and coauthor of The XML Handbook, thinks of it this way: A telephone book has certain data elements--names and phone numbers listed in alphabetical order--and because of that, you'd recognize a phone book by a single page, regardless of its title or binding. In other words, the "phone book" category (or "document type," in XML-speak) is defined by the fact that it contains names and phone numbers (the "elements"). If a book contained maps, say, or embedded the phone numbers within a series of short stories, then we probably wouldn't call it a phone book.

So it is with XML documents. The DTD sets the rules that make elements legal (which is where those elements can appear within the document), and they define how each element is displayed on-screen. The DTD is then linked to an XML document, either by reference or by embedding it within the XML document itself. (Embedded DTDs tend to be shorter than external DTDs, and since they are read first, they can be used to override portions of the external DTD. This can be handy if you want to apply specific rules to certain documents.)

Further, an XML document must conform to certain grammatical rules. Two standards apply. A "well-formed" document follows the specific XML rules for tags. A "valid" document follows the rules set out in your DTD.

You don't need a DTD if your document is well formed, which is to say that it conforms to the rules of the official XML specification. (The spec is at www.w3.org/TR/REC-xml.) Just remember that the syntax is fussier than HTML. For example, every tag must be accompanied by an end tag: You might write <paragraph>This is a list of customer cards.</paragraph>. Tags are also case sensitive: <president>John Doe</president> is not the same as <PRESIDENT>John Doe</ PRESIDENT>. However, if your XML document is well formed, a browser can read it on the fly, just as it can read an HTML document.

So why bother with a DTD at all? Because a DTD will let you actually process data, which, after all, is the whole point behind XML. If you only want to display static information on a Web page, you probably won't need to write a DTD. However, if you want to identify specific elements (like, say, credit card numbers) and manipulate data, you have to declare which elements are which and where they appear within the document. That's the DTD's job.

The Do-it-yourself dtd

Writing a DTD doesn't have to be difficult. The first thing you'll have to do is decide what kind of document you want to create. Are you writing Web pages? Are you linking to a database? Each document type is structured differently and contains different kinds of information.

Consider this sample DTD, from Richard Lander of the University of Waterloo in Canada, which outlines the structure of a published novel.

<!ELEMENT novel (preface,chapter+,biography?,criticalessay*)>

<!ELEMENT preface (paragraph+)>

<!ELEMENT chapter (title,paragraph+,section+)>

<!ELEMENT section (title,paragraph+)>

<!ELEMENT biography (title,paragraph+)>

<!ELEMENT criticalessay (title,section+)>

<!ELEMENT paragraph (#PCDATA)>

<!ELEMENT title (#PCDATA)>

In this example, the basic setup and sequence of the book is laid out, and rules are established for how often each element can (or must) occur. The plus signs indicate that an element is required and can occur in multiple instances. For ex ample, "chapter+" in the first line of code (where the basic novel structure is established) means that the novel must contain at least one chapter, but there are no limits on how many chapters you can have. The asterisks designate elements that are optional but can occur more than once--"criticalessay*" in the opening line means you can have more than one critical essay, but you're not required to have any. The question marks indicate that the element must occur once or not at all. If an element has none of these three signs, it must appear only once. The first line of our DTD shows "preface" without a plus sign, question mark, or asterisk, so you must have only one preface.

The "#PCDATA" content in the last two lines of the DTD determine where character data can be placed. In the example, text can be placed in paragraphs and titles, but not directly in prefaces or chapters.

The Easy Way Out

Are you bewildered yet? Don't be. Thousands of DTDs already exist in disciplines ranging from education to chemistry to math, and they can be downloaded from the Web. Check out www.schema.net for some examples.

If you insist on doing it yourself, a good authoring package will spare you the hardship of coding your DTD by hand. For example, Microstar's Near & Far Designer 3.0 automates the DTD creation process and lets you modify DTDs visually. Your data appears in a tree structure, and you can define which elements you want to include, in sequence and quantity. Several free authoring programs are also listed at www.oasis-open.org/cover/ and xml.com/xml/pub/pt/Authoring. If you do just a little research, you should be able to find a program that suits your needs.

An author, online specialist, and contributing editor, Elizabeth Crowe also writes the Net Surfer column for Computer Currents.

XML on the Web

Confounded, confused, and otherwise mystified by XML? Fear not. There are plenty of Web sites to help you get your bearings.

pdbeam.uwaterloo.ca/~rlander/

This site, hosted by the University of Waterloo in Waterloo, Canada, offers lots of background information on XML. The "Tutorial in XML and XSL Authoring" is especially useful and will walk you through the steps of creating an XML document.

www.oasis-open.org/cover/sgml-xml.html

Can't get enough XML? Try Robin Cover's SGML/XML Web page, which contains a gigantic bibliography with over 2,000 entries. You can even link to the personal home pages of XML experts. The news stories are interesting, too, and very timely.

www.sgmlsource.com

Although it's mostly concerned with SGML, the personal home page of Charles Goldfarb, the father of SGML, is notable for his reviews of XML books. If you're going book shopping, check here first.

www.textuality.com/xml/

Tim Bray is one of the more knowledgeable XML pundits around, and he has put together this handy primer that seems to be aimed mostly at neophytes. The FAQ is noteworthy for its concise, matter-of-fact answers, and you can even take a look at Lark, Bray's XML parser.

www.ucc.ie/xml/

Maintained by the W3C's Special Interest Group, this page is basically an FAQ divided into four sections, depending on whether you're a user, author, developer, or just a curious Webhead looking for general information. Caution: The site features links aplenty, so if you're the type of surfer who gets lost in wild goose chases, put your blinders on.

www.w3.org/XML/

The World Wide Web Consortium hosts this site, which keeps you posted on the W3C's XML activities. All the W3C requirements and recommendations are listed here, along with the actual XML specification and a calendar of upcoming events. Check out the Activity Statement for a rundown of what the W3C has up its sleeve.

www.xml.com

This is one of the better XML pages and is a good place to keep tabs on the world of XML. Here you'll find articles on how different organizations are using XML, a nifty "What Is XML?" primer, hints on topics such as converting a DTD from SGML to XML, and a hefty resource guide.

www.xmlx.com

This online forum is a clearinghouse for sharing DTDs in disciplines ranging from health care to business productivity to the military. If you're looking for editors, parsers, or browsers, try the Software link.