Tutorial: Introduction to XML

Doug Tidwell
Cyber Evangelist, XML developerWorks Team
Raleigh, NC
July, 1999

About this Tutorial

XML, the Extensible Markup Language, has been hailed as a technical revolution comparable in scope to sliced bread and the wheel. This tutorial cuts through the hype to show you what XML is all about. XML is an important new technology that will revolutionize the web. Just as important, XML **isn't** rocket science. There's no reason you can't start using XML to transform your business today.

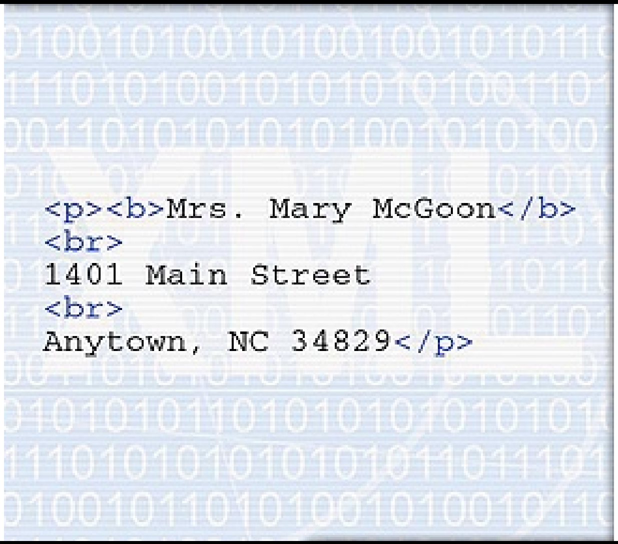
About the Author

Doug Tidwell is a Senior Programmer at IBM. He has well over a tenth of a century of programming experience and has been working with XML-like applications for several years. His job as a Cyber Evangelist is to help customers evaluate and implement XML technology. He holds, with special gloves, a Masters Degree in Computer Science from Vanderbilt University and a Bachelors Degree in English from the University of Georgia.

Section 1 - What is XML?

What is XML?

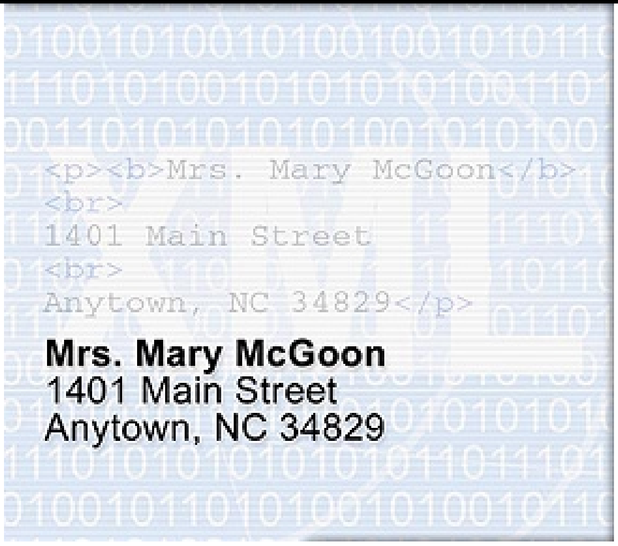
Extensible Markup Language, or XML for short, is a new technology for web applications. XML is a World Wide Web Consortium standard that lets you create your own tags. XML simplifies business-to-business transactions on the web.



```
<p><b>Mrs. Mary McGoon</b>
<br>
1401 Main Street
<br>
Anytown, NC 34829</p>
```

Why Do We Need XML?

Why do we need XML when everyone's browser supports HTML today? To answer this question, look at the sample HTML Code shown. HTML tags are for browsing; they're meant for interactions between humans and computers.

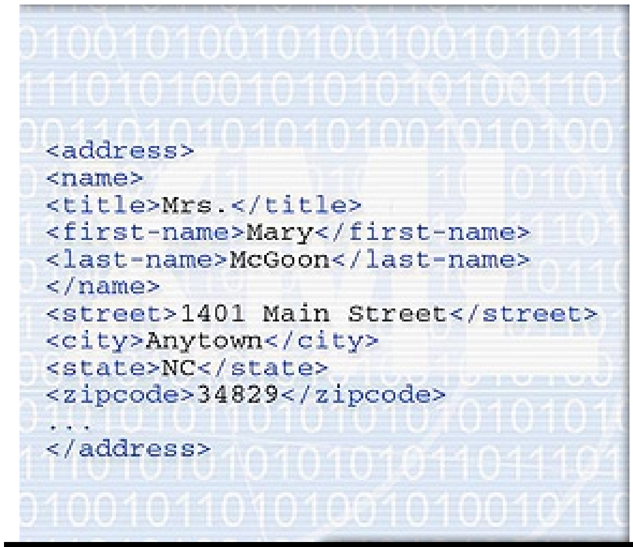


```
<p><b>Mrs. Mary McGoon</b>
<br>
1401 Main Street
<br>
Anytown, NC 34829</p>
```

Mrs. Mary McGoon
1401 Main Street
Anytown, NC 34829

Rendering HTML

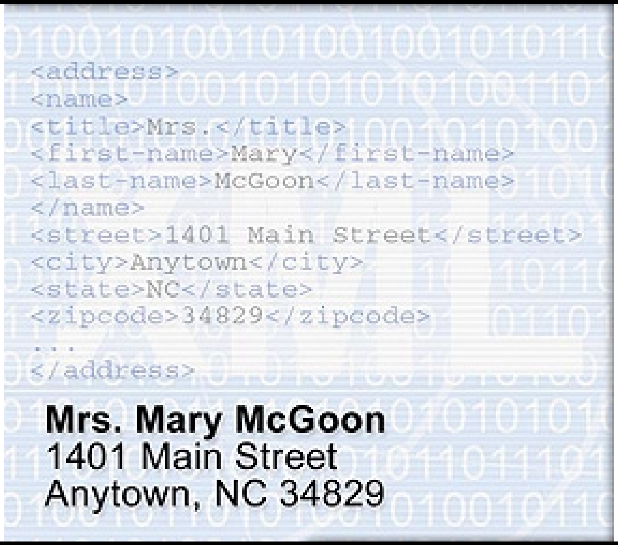
When rendered, the HTML in the previous example looks like this. As you can see, HTML tags describe how something should render. They don't contain any information about what the data is, they only describe how it should look.



```
<address>
<name>
<title>Mrs.</title>
<first-name>Mary</first-name>
<last-name>McGoon</last-name>
</name>
<street>1401 Main Street</street>
<city>Anytown</city>
<state>NC</state>
<zipcode>34829</zipcode>
...
</address>
```

Sample XML Code

Now let's look at some sample XML Code. With XML, you can understand the meaning of the tags. More importantly, a computer can understand them as well. It's easier for a computer to understand that the tag `<zipcode>34829</zipcode>` is a zip code.

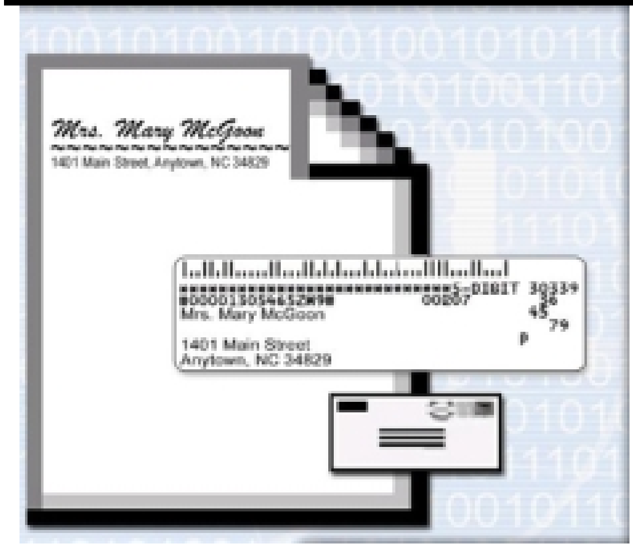


```
<address>
<name>
<title>Mrs.</title>
<first-name>Mary</first-name>
<last-name>McGoon</last-name>
</name>
<street>1401 Main Street</street>
<city>Anytown</city>
<state>NC</state>
<zipcode>34829</zipcode>
...
</address>
```

Mrs. Mary McGoon
1401 Main Street
Anytown, NC 34829

Rendering XML

XML from the previous example might be rendered like this. Notice that even though the tags are different, they can still be rendered just like HTML.

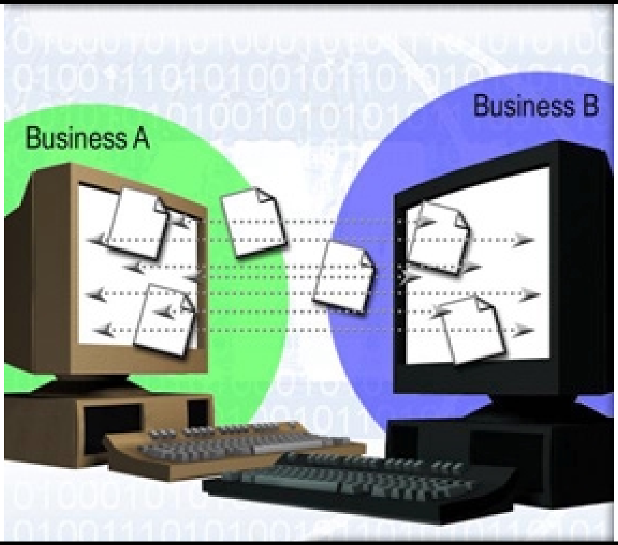


A Second Rendering of XML

By applying a different stylesheet to the same document, an XML document can be rendered in different formats. The key is that with XML the information is in the document, while the rendering instructions are elsewhere. In other words, content and presentation are separate.

How XML Will Change the Web

XML applications provide many advantages. Let's talk about several of those now.
XML's strongest point is its ability to do data interchange. Because different organizations (or even different parts of the same organization) rarely standardize on a single set of tools, it takes a significant amount of work for two groups to communicate. XML makes it easy to send structured data across the web so that nothing gets lost in translation.



Enable Business to Business Communication

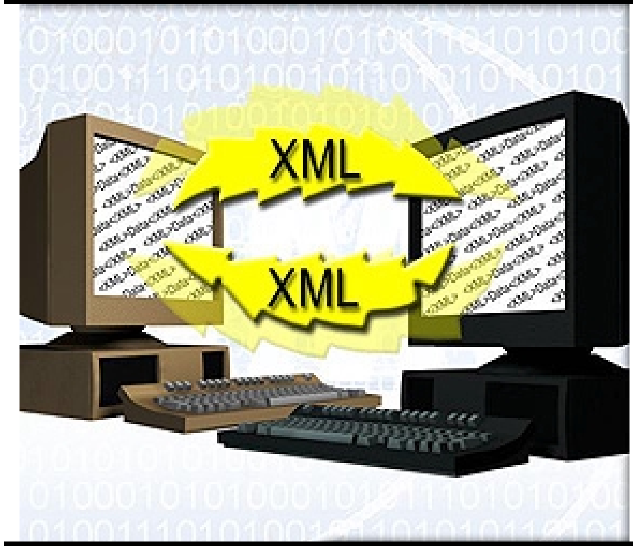
When using XML, I can receive XML-tagged data from your system, and you can receive XML-tagged data from mine. Neither of us has to know how the other's system is organized. If another partner or supplier teams up with my organization, I don't have to write code to exchange data with their system. I simply require them to follow the document rules defined in the DTD.

Enable Smart Agents

When writing an agent, one of the challenges is to make sense of incoming data. A good agent interprets information intelligently, then responds to it accordingly. If the data sent to an agent is structured with XML, it's much easier for the agent to understand exactly what the data means and how it relates to other pieces of data it may already know.

Enable Smart Searches

One major problem with today's web is that search engines can't process HTML intelligently. For example, if you search on "chip" when you're looking for someone named Chip, you might get pages on chocolate chips, computer chips, and guys named Chip. But if there were a DTD for name and address records, searching for a guy named Chip could generate much more accurate and useful search results.



Summary

So why do we need XML? XML makes it easier for two computers to exchange data with each other. Your data is described using tags that describe what each piece of data is. XML doesn't replace HTML, though; they're designed for different purposes. XML is the Web's language for data interchange and HTML is the Web's language for rendering.

Review

Let's review. When is it appropriate to use XML?

A. When I want to display something in a browser.

B. When I feel like creating my own tags.

C. When I want to add a buzzword to my resume.

D. When I need to send self-describing data to another machine or application.

[The correct answer is D.]

Section 2 – How can I use XML today?

Accessing XML on the Client

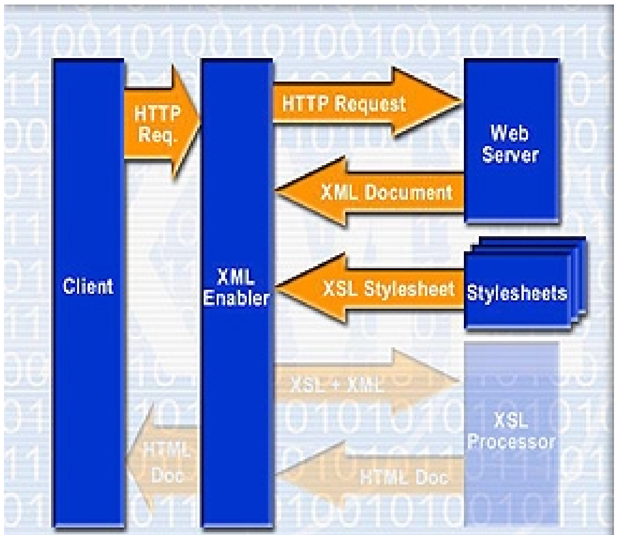
XML changes the way data moves across networks. XML encapsulates data inside custom tags that carry semantic information about the data. A common question is, "Once I've got some XML-tagged data, how do I view it?"

IBM Solutions

The answer to that question is the focus of this topic. IBM has solutions that let you view XML data on any client or browser. IBM's technology allows you to take full advantage of the capabilities of each platform; you don't have to take a lowest common denominator approach. If your client or browser supports XML or Dynamic HTML, you can use those features, while still generating useful views of your XML-tagged data for low-function clients.

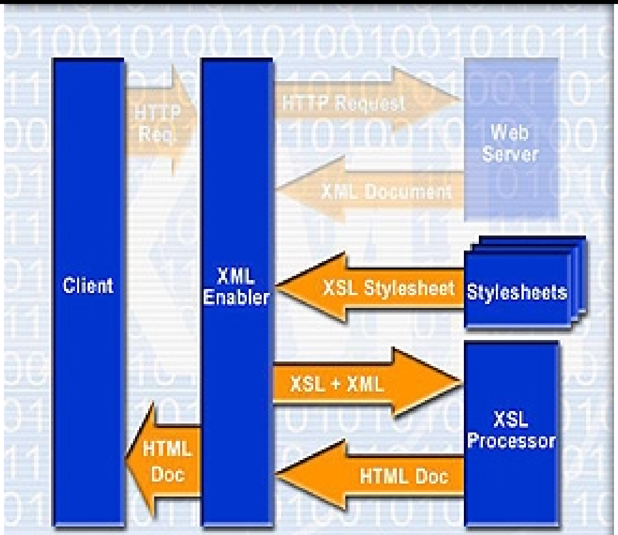
The XML Enabler

One IBM solution for doing this is the XML Enabler, available on IBM's alphaWorks site. It allows you to convert XML-tagged data into HTML, using different style sheets for different browsers. We'll discuss the XML Enabler briefly, then we'll show you how to view XML-tagged data on any browser you choose.



The XML Enabler

The architecture of the XML Enabler is simple. When a browser attempts to load an XML document, the XML Enabler looks at the type of browser that requested the document. The XML Enabler then selects a stylesheet written in the Extensible Stylesheet Language, or XSL, based on the browser type.

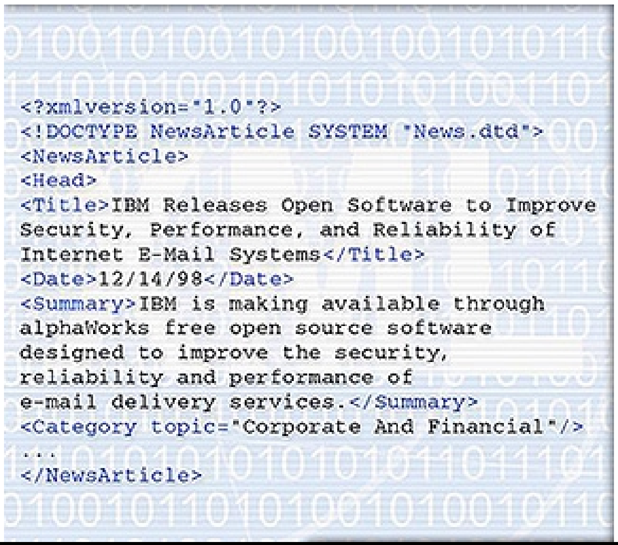


The XML Enabler

The XML Enabler retrieves the requested XML document, processes it with the appropriate stylesheet, and returns the results to the browser. This allows you to generate different views of an XML document, and each view takes advantage of as many browser features as possible.

Implementation

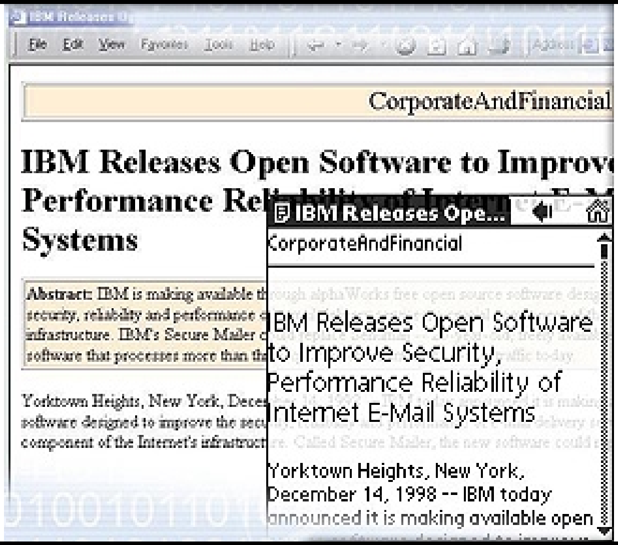
The XML Enabler is implemented as a Java servlet. It is built on the XML4J parser from IBM and the Lotus XSL Processor, both of which are available on IBM's alphaWorks site. All of the code is written in Java, so it's platform independent.



```
<?xmlversion="1.0"?>
<!DOCTYPE NewsArticle SYSTEM "News.dtd">
<NewsArticle>
<Head>
<Title>IBM Releases Open Software to Improve
Security, Performance, and Reliability of
Internet E-Mail Systems</Title>
<Date>12/14/98</Date>
<Summary>IBM is making available through
alphaWorks free open source software
designed to improve the security,
reliability and performance of
e-mail delivery services.</Summary>
<Category topic="Corporate And Financial"/>
...
</NewsArticle>
```

A Sample XML Document

Before we demonstrate viewing XML on various clients, let's look at a sample XML document. Our sample document is a news item containing a title, a summary, a category, and other pieces of information.



CorporateAndFinancial

IBM Releases Open Software to Improve Performance Reliability of Internet E-Mail Systems

Abstract: IBM is making available through alphaWorks free open source software designed to improve the security, reliability and performance of ... infrastructure. IBM's Secure Mailer ... software that processes more than ... traffic today.

Yorktown Heights, New York, December 14, 1998 -- IBM today announced it is making available ... software designed to improve the security ... delivery ... component of the Internet's infrastructure. Called Secure Mailer, the new software could ...

CorporateAndFinancial

IBM Releases Open Software to Improve Security, Performance Reliability of Internet E-Mail Systems

Yorktown Heights, New York, December 14, 1998 -- IBM today announced it is making available open

A Rendered XML Document

These two screen captures illustrate the range of browsers the XML Enabler can support. In the background, Microsoft's Internet Explorer Version 5 renders an XML document using an XSL stylesheet. In the foreground, a palmtop browser renders an HTML version of the same XML document. The XML Enabler determines what kind of document is delivered to each browser.

alphaWorks and developerWorks

IBM's alphaWorks site at www.alphaWorks.ibm.com contains many XML-related tools and technologies, including parsers, editors, and stylesheet processors.

In addition, the XML Zone of IBM's developerWorks site, www.ibm.com/developer, contains several programming tutorials related to XML.

Summary

In summary, the XML Enabler allows you to begin working with XML technology today. You can serve XML-tagged data to any type of client without having to change your existing systems or browsers. If your users have modern browsers, you can create XSL stylesheets that use their latest features. At the same time, you can support users with lower-function browsers. Best of all, the XML Enabler lets you do these things in a way that's easy to support and expand as your environment changes.

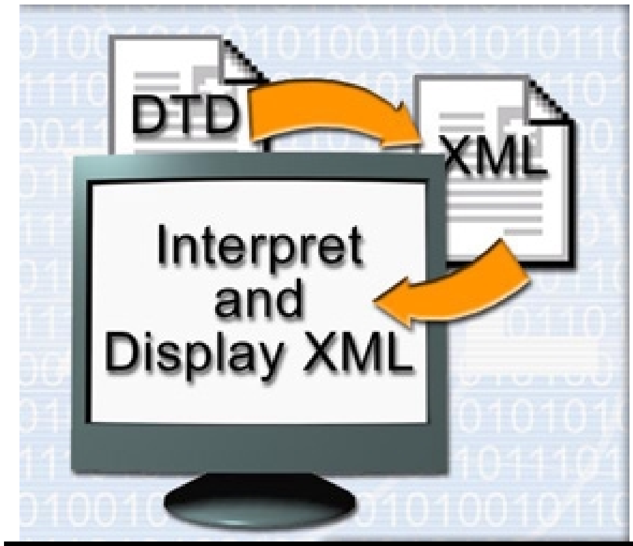
Review

Let's review. From a business perspective, which of the following statements about XML is true?

A. There's no point migrating to XML until your whole enterprise has installed Internet Explorer V5.
B. We'll migrate to XML when most of the major browser vendors support it.
C. We can use XML documents here and there, but we'll have to convert them to a single text-only view that all browsers can handle.
D. We can use XML today and transform our XML documents so that any browser can render them attractively and usefully.

[The correct answer is D.]

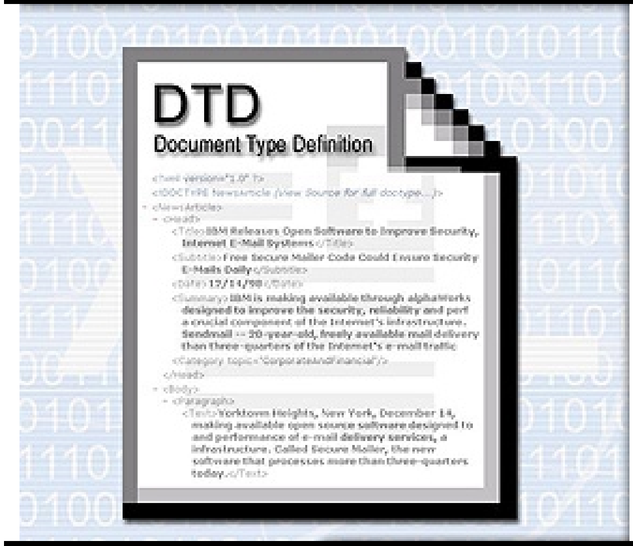
Section 3 – Applying XML



Introduction

In order to build XML applications, you will need to do four things:

- Select or write a DTD
- Generate XML documents
- Interpret XML documents, and
- Display XML documents.



```
<?xml version="1.0" ?>
<!DOCTYPE NewsArticle (View Source for full doctype...)>
- <NewsArticle>
  - <Head>
      <Title>IBM Releases Open Software to Improve Security,
        Internet E-Mail Systems</Title>
      <Subtitle>Free Secure Mailer Code Could Ensure Security
        E-Mails Daily</Subtitle>
      <Date>12/14/98</Date>
      <Summary>IBM is making available through alphaWorks
        designed to improve the security, reliability and perf
        a crucial component of the Internet's infrastructure.
        Sendmail -- 20-year-old, freely available mail delivery
        than three-quarters of the Internet's e-mail traffic
      <Category topic="CorporateAndFinancial"/>
    </Head>
  - <Body>
    - <Paragraph>
        <Text>Yorktown Heights, New York, December 14,
          making available open source software designed to
          and performance of e-mail delivery services, a
          infrastructure. Called Secure Mailer, the new
          software that processes more than three-quarters
          today.</Text>
```

Selecting or Writing a DTD

What is a DTD? A document type definition defines what tags can go in your document, what tags can contain other tags, the number and sequence of the tags, the attributes your tags can have, and optionally, the values those attributes can have.

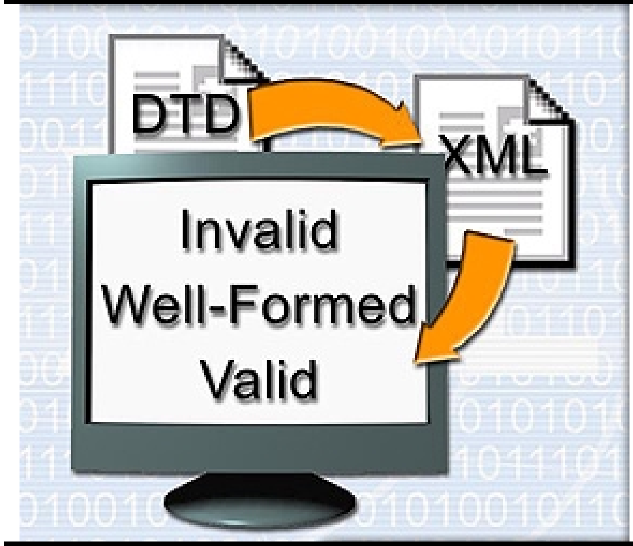
Generating an XML Document

Think of the DTD as a template you fill in. For each element, get the data from its location such as a database query, a full-text search, or a directory query, and then simply put it in the template. When the template is full (all the required elements are filled in and the document is valid), you can send it to the requestor of the document.

Tag Rules for XML Documents

There are a couple of rules for XML tags that are different from HTML tags:

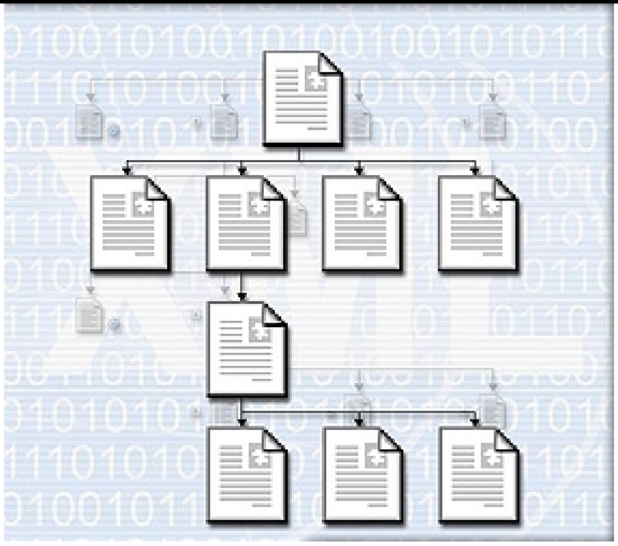
- **XML tags can't overlap.** `<a><b></a></b>` isn't allowed. If you start a `<b>` tag inside an `<a>` tag, you have to end it inside the `<a>` tag as well.
- **You can't leave out any end tags.** Tags like `</p>` and `</br>` are required.
- **Tags that don't contain any text can contain the end marker at the end of the start tag.** In other words, `<br></br>` is equivalent to `<br />`.



Invalid, Well-Formed, and Valid Documents

There are three kinds of XML documents:

- **Invalid Documents:** Documents that don't follow the tag rules we just discussed. If a document has a DTD, and it doesn't follow the rules defined in its DTD, that document is invalid as well.
- **Well-Formed Documents:** Documents that follow the XML tag rules, but don't have a DTD.
- **Valid Documents:** Documents that follow both the XML tag rules and the rules defined in their DTDs.



Interpreting XML Documents

When you need to interpret an XML document, there are two APIs you can use: The Document Object Model, or DOM, and the Simple API for XML, or SAX.

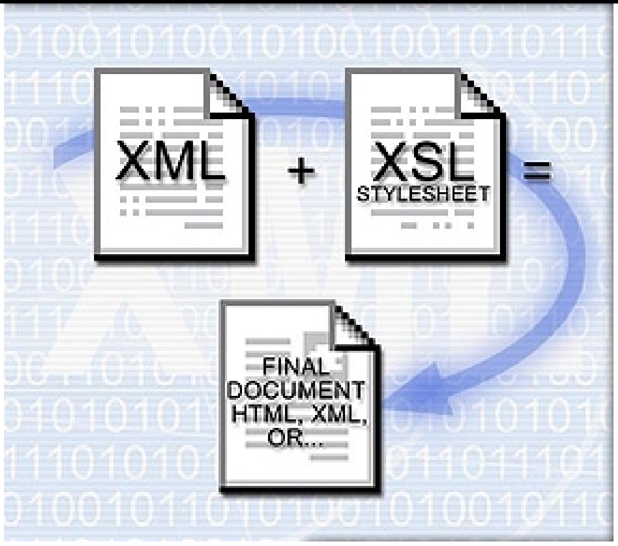
The DOM is a standard of the World Wide Web Consortium that creates a tree view of your XML document. The DOM provides standard functions for manipulating the elements in your document.

Interpreting XML Documents

The SAX API notifies you when certain events happen as it parses your document. When you respond to an event, any data you don't specifically store is discarded.

SAX or DOM?

Why would you use SAX or DOM? If your document is very large, using SAX will save significant amounts of memory when compared to using DOM. This is especially true if you only need a few elements in a large document. On the other hand, the rich set of standard functions provided by the DOM isn't available when you use SAX.



Displaying XML Documents

There are several ways you can display XML documents. If your browser can display XML, you can simply send the document out to the browser. Or use an XSL stylesheet to transform the XML into something your browser can handle. An XSL stylesheet contains some number of templates that define how the elements in an XML document should be transformed.

Displaying XML Documents

If you want to do complicated sorting or restructuring that's beyond the realm of XSL, use DOM. In this method, you parse the XML document, then write Java code to manipulate the DOM tree in whatever way you wish. Your code has complete access to the DOM and all of its methods, so you're not bound by the limitations or design decisions of XSL.

Review

We've covered a lot! Let's review. Which of the following is **not** a way to display an XML document?

A. Using the DOM.

B. Using an XML-capable browser.

C. Using expensive, proprietary software than can only be bought from certain vendors.

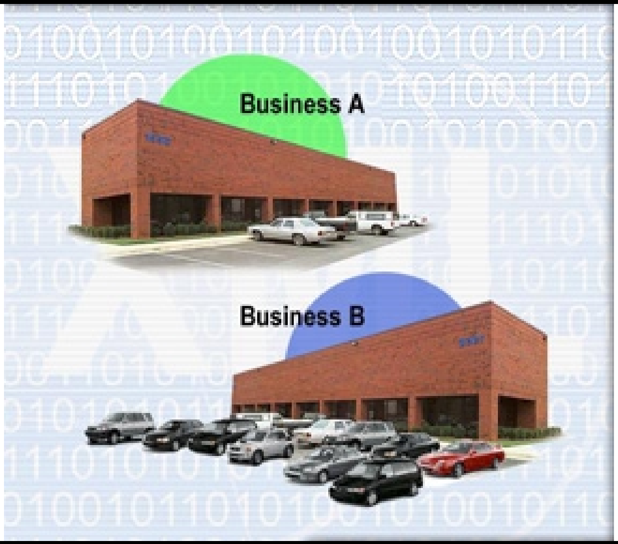
D. Using XSL.

[The correct answer is C.]

Section 4 – Case Study

CEO Overview

Hello, my name is Amanda Reckonwith, president and CEO of Hard to Find Auto Parts, a multinational conglomerate. Our mission is to provide our customers with high quality auto parts in a timely manner and at a fair price. The secret to our success is our ability to locate hard to find auto parts from our suppliers over the internet, as well as our ability to show up as a parts source when our customers conduct internet searches.



Turning to XML

Due to increasing competition in the marketplace, we must be able to find parts quickly and at the best available price. We've turned to XML to conduct e-business within our marketplace. We have two simple choices. We can either conform to an emerging standard, and even help define it, or get left behind. Let's talk to John, our XML developer!

John, XML Developer

Hello, my name is John. I'm an XML application developer here at Hard to Find Auto Parts. I'm responsible for ensuring that we meet our business challenges. I'll describe to you our rationale for using XML.

XML makes searches smarter. As an example, consider the extranet that we share with our suppliers. When we need to search the extranet to find a part, XML allows us to find exactly what we're looking for. We share a DTD with all of our suppliers, so our searches return exactly what we're looking for.

Make Searches Smarter

Before we used XML, our searches returned thousands of hits, most of which were garbage. If we looked for a window for a Honda Accord, we might get a news article about the Middle East Peace Accord. With an XML-based search, all our results are XML documents that contain a `<model>Accord</model>` tag.

Maintain Communications

The fact that we use XML to represent our data on the Internet and within our extranet means that our suppliers and customers are insulated from any changes we might make to our infrastructure. We're insulated from any changes in their systems as well. Because data interchange is one of XML's strengths, we're able to move data across different systems without losing anything in translation.

Smart Agents

Because all of our suppliers send data about their inventory and prices in XML, we've been able to write agents that process that data intelligently. For example, we have agents that watch certain auto parts in our inventory control system. When only a few of a particular item are in stock, our agent checks with all our vendors to find out which one has the part in stock and has the best price. Using XML has given us a clear advantage over our competitors.

Customer Recognition

As the leading XML-based auto parts dealer, information from our database turns up on XML search engines all over the world. Customers who are tired of bogus HTML searches can use our XML-based approach to find exactly what they're looking for. Auto parts houses that haven't caught on to XML don't have a chance to make a sale to these customers.

Review

Ready for a review? Which of the following is **not** an advantage of using XML over standard markup languages?

A. Using XML provides us with the ability to conduct smarter searches.

B. Using XML provides better business to business communication.

C. XML incorporates the use of agents that process data intelligently.

D. XML tags are more difficult to learn than standard HTML or DHTML tags.

[The correct answer is D.]