



IBM IT Education Services

Session ID: E14

**Speaker : Janice Winchell
MICOM Inc**

XML ... Pretty **HOT** stuff

November 10 - 12, 2003 | Hilton, Las Vegas, NV

© 2003 IBM Corporation

XML - pretty *HOT* stuff

THE FOLLOWING TERMS ARE TRADEMARKS and/or COPYRIGHTS OF INTERNATIONAL BUSINESS MACHINES CORPORATION:

COBOL/370	MLE
COBOL for VSE/ESA	Millennium Language Extensions
COBOL for MVS & VM	DB2
COBOL OS/390	VisualAge COBOL
LANGUAGE ENVIRONMENT/370	
LANGUAGE ENVIRONMENT for MVS & VM	
LANGUAGE ENVIRONMENT OS/390	
LANGUAGE ENVIRONMENT for VSE/ESA	
MVS	
MVS/ESA	
OS/390	
VSE/ESA	
IBM	
CICS CICS/ESA	
SYSTEM/390	

XML is a trademark of the Massachusetts Institute of Technology

Internet Explorer is a trademark of Microsoft Corporation

The information presented in this material is for illustration only. There is no implied warranty or correctness for the usability or applicability of the material contained herein. Examples of programming techniques are for demonstration of language elements, and may not necessarily be applicable to specific programming requirements.

Introduction

to

XML

XML - pretty *HOT* stuff

Introduction to XML

XHTML as well

HTML can do a lot, but XML can do more

What are the parts of an XML document?

What is the syntax (grammar) that XML expects?

More things to understand

Document Type Definitions (DTD)

XML namespaces

XML schemas

XSL - extensible style sheets

XML Linking

Document Object Model (DOM)

Creating XML documents

Elements, Attributes, Entites

Where to find out more information

Additional resources

..OK....fasten your seat belts - XML is going to be an exciting ride!

The most visible definition of XML comes from the XML Specification V1 of February 1998

“The eXtensible Markup Language (XML) is a subset of SGML (Standard Generalized Markup Language) that is completely described in this document. Its goal is to enable generic SGML to be served, received and processed on the Web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML”

Or, you can say XML *is* SGML

“XML is an application profile or restricted form of SGML, the Standard Generalized Markup Language (ISO 8879). By construction, XML documents are conforming SGML documents.

XML - pretty *HOT* stuff

Design goals for XML

XML - straightforward and usable over the internet

Not just over the internet, but also within the enterprise

XML - support wide variety of applications

Beyond tools that live on Web servers

XML - compatible with SGML

Have to keep enough SGML for interoperability

Make it easy to process XML documents

Keep optional features to absolute minimum, ideally zero

SGML options make for flexibility and power, but also make for complication

XML documents - need to be human-legible and reasonably clear

Allows use of text editors

XML design - prepared quickly

Didn't want to spend TOO long in development creating another browser war

Design of XML - formal and concise

Makes language programmer friendly

XML documents - easy to create

Allows wide variety of creation tools for many varied users

Terseness in XML markup minimally important

Avoid complexities of SGML

Choose clarity over cleverness

XML - pretty *HOT* stuff

XML and its relations

....XML is not “one technology” - instead a *FAMILY of related standards*

- **eXtensible Style Language (XSL)**
 - Similar to Cascading Style Sheets (CSS) working with HTML
- **eXtensible Hypertext Markup Language (XHTML)**
 - Deliver on different media types (palm, etc)
- **XML Linking Language**
 - **Xpath**
 - Perform actual addressing of parts rather than entire XML document
 - **Xlink**
 - Use XML syntax to define relationship between 2 or more data objects or portions of objects (not just entire document)
 - **Xpointer**
 - Builds on XPATH for internal addressing using XML markup to link
- **XML Namespaces**
 - Allow assigning unique name to document constructs (avoid duplicate names)
- **XML Schemas**
 - Allow constrains and rules
- **Document Object Model (DOM)**
 - Allow language neutral application interface to manipulate HTML and XML data
- **Scalable Vector Graphics (SVG)**
 - XML application for rendering 2-dimensional graphic objects

Why use XML?

- **Structured data**
 - Can extract just the information necessary
- **Data exchange**
 - Allows exchange of database contents
 - Can send information in a structure that can be parsed (you define the tags)
- **XML complements HTML**
 - XML data can be used in HTML page
- **Self-describing**
 - No prior knowledge of an application is needed
- **Search engines**
 - Search relevancy increases due to contextual information in XML document
- **Updates**
 - Document Object Model built into XML allows access and update of individual elements (not required to update entire site page by page)
- **User-selected view of data**
 - Different users, different media can access different information or present information in different ways

Why use XML?

- **Can replace proprietary tagging systems with platform independent meaningful tags**
 - Applications can now parse this information

```
<!--Price List for veggies -->
<d1>
  <!-- Veggie -->
  <dt> Broccoli</dt>
  <!-- Price -->
  <dd>$1</dd>
  <!-- Veggie -->
  <dt>Cauliflower</dt>
  <!-- Price -->
  <dd>$2</dd>
</d1>
```

now can look like:

```
<VeggiePrice>
  <Veggie>Broccoli</Veggie>
  <Price>$1</Price>
  <Veggie>Cauliflower</Veggie>
  <Price>$2</Price>
</VeggiePrice>
```

XML - pretty *HOT* stuff

Tools for XML - You can build it yourself “by hand” with text editor

Not a bad idea if you are just starting

1. and not a bad way to get the “practice” and understanding
- Make the XML Declaration (not required but recommended to have one)
 - `<?xml version="1.0" standalone="no" encoding="UTF-8"?>`
 - Create root element
 - You pick the element “name”
 - Create XML Code
 - Tags have start and end components
 - Tags are nested correctly
 - Attribute values must have quote bounds (either “ or ‘)
 - Empty tags must be correctly formatted (use a / before the closing >)
 - Check the coding (these are just a couple of places)
 - IBM’s can be found by going to [www.ibm.com](http://www.ibm.com/software/webservers/appserv/doc/v30/ae/web/apidocs/com.ibm....), searching on XML Validator and picking one that starts out so:
<http://www- .ibm.com/software/webservers/appserv/doc/v30/ae/web/apidocs/com.ibm....>
 - the actual url is way long!
 - Microsoft’s XML Validation page
 - http://msdn.microsoft.com/downloads/samples/internet/xml/xml_validator

XML - pretty *HOT* stuff

Tools for XML

The screenshot shows a Microsoft Internet Explorer browser window. The title bar reads "XML Matters: Roundup of XML editors, Part 1 - Microsoft Internet Explorer provided by AT&T WorldNet Service". The address bar shows the URL "http://www-106.ibm.com/developerworks/xml/library/x-matters21/". The page content includes the IBM logo, a search bar, and navigation links. The main article title is "XML Matters: Roundup of XML editors, Part 1" with a sub-header "Revisited products for Java and MacOS". The author is David Mertz, Ph.D. The article text discusses the progress of XML editors and lists several products: Morphon Technologies' Morphon 2.0.5, SyncRO's $\langle \text{oxygen} \rangle$ 1.2.1, ElfData's XML Editor 1.14, Altova's XML Spy 4.4, Wattle Software's XMLWriter, NetBryx Technologies' EditML Pro 2.6, and Corel XMetal 3. A right-hand sidebar contains a "Contents" section with links to various parts of the article and a "Related content" section with links to other developerWorks articles.

XML Matters: Roundup of XML editors, Part 1

Revisited products for Java and MacOS

Level: Introductory

David Mertz, Ph.D. (mertz@gnosis.co)

Transformer, Gnosis Software, Inc.

August 1, 2002

In this two-part series on XML editors, David looks at the progress of commercial tools in the year-and-a-half since he last looked at this tools category. These tools have progressed from largely cosmetic wrappers around text editors to fleshed-out development environments that substantially ease the process of working with XML-oriented technologies. This first installment examines Java and MacOS applications, specifically Morphon Technologies' Morphon 2.0.5, SyncRO's $\langle \text{oxygen} \rangle$ 1.2.1, and ElfData's XML Editor 1.14.

Since the last time (see [Resources](#)) I looked at XML editors, quite a lot has changed. A great deal of progress has been made, and in this two-part series you'll get the details on the new features and offerings of some of the editors available to you. Of the nine editors I set out to review in this series, I was unable to obtain two -- XMLmind XML Editor (XXE) and Vervet Logic's XML Pro. The remaining seven reviews are split over two installments to allow me to look in detail at the features of each product. In this, the first part, I look at the tools targeted at JVMs and the MacOS (or, from my perspective -- programs I can run on my iBook). In researching products, I didn't find anything Linux/Unix specific that was both current and of similar sophistication to those XML editors I'm reviewing here. Of course, the Java-based tools will run fine under Linux, as well as on other Java-enabled platforms.

I am deliberately leaving out any discussion of general text editors, including everything-but-the-kitchen-sink editors like (X)Emacs and more modest but customizable personal favorite text editors.

The products that I'll cover in this roundup are:

- Morphon Technologies' Morphon 2.0.5
- SyncRO's $\langle \text{oxygen} \rangle$ 1.2.1
- ElfData's XML Editor 1.14
- Altova's XML Spy 4.4
- Wattle Software's XMLWriter
- NetBryx Technologies' EditML Pro 2.6.
- Corel XMetal 3

Contents:

- [What makes a good editor?](#)
- [Morphon Technologies' Morphon 2.0.5](#)
- [SyncRO's $\langle \text{oxygen} \rangle$ 1.2.1](#)
- [ElfData's XML Editor 1.14](#)
- [Other players](#)
- [Summary](#)
- [Resources](#)
- [About the author](#)
- [Rate this article](#)

Related content:

- [Roundup of XML editors, Part 2](#)
- [XML Matters: A roundup of editors](#)
- [Subscribe to the developerWorks newsletter](#)
- [developerWorks Toolbox subscription](#)

Also in the XML zone:

- [Tutorials](#)
- [Tools and products](#)
- [Code and components](#)
- [Articles](#)

XML - pretty *HOT* stuff

Tools for XML

The screenshot shows a Microsoft Internet Explorer browser window. The title bar reads "XML: In search of a good editor - Microsoft Internet Explorer provided by AT&T WorldNet Service". The address bar shows "http://www.ivritype.com/xml/". The page content includes a search box, a list of XML editors, and an introduction section.

XML Editors: Allegations of Functionality in search of reality

Search Ivritype for:

Editors explored

- [Clip 1.5](#)
- [Homesite 4.0](#)
- [Visual XML 1.1b](#)
- [Xeena 1.03](#)
- [XMetaL 1.0](#)
- [XML Pro 2.0](#)
- [XML Spy 2.5](#)
- [Conclusions](#)
- [Recommended](#)

Introduction

The term, "XML editor" can refer to a great many types of tools, depending on the purpose to which the editor is to be put. Some criteria are described in this ZD Net overview: <http://www.zdnet.com/devhead/stories/articles/0,4413,2138258,00.html>

For a quick overview of most available XML editors, see <http://www.xmlsoftware.com/editors/> (If you browse a bit, you'll also see some fuzzy categories on the site, which accurately reflect the current state of art.)

I have gone through several sets of criteria in reviewing software. In the end, this was the minimal, "this is what we really need":

- The tool must be able to open up multiple documents
- Find/Replace, including global, search a directory find/replace
- It must show the editor relevant tags and make tagging easy
- It must validate the results and make it easy to find errors.
- It must be able to switch, update, or add DTDs later.
- It must allow the creation and/or editing of well-formed XML documents sans DTD.
- If the tool can accommodate at least some types of external

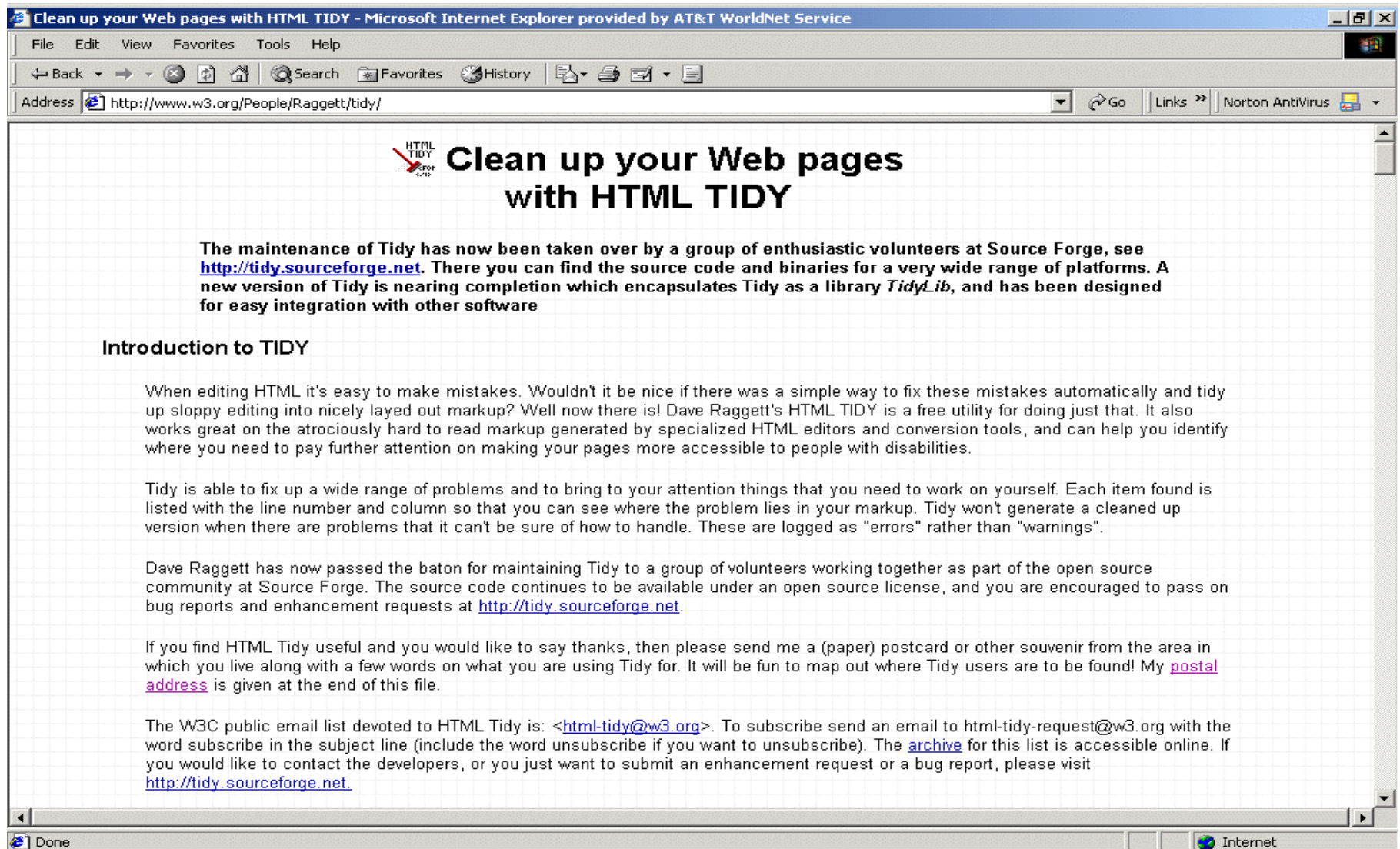
Done Internet

XML - pretty *HOT* stuff

- XHTML - intended to provide easy transition between HTML and XML
 - Extensible Hypertext Markup Language
 - HTML like markup language
 - Users can extend the language
 - XHTML is defined as an XML DTD
 - HTML is defined as an SGML DTD
 - Designed to address the issue of how to re-use existing HTML
 - Retain compatibility with existing browsers
 - Extend with new features for XML-enabled
 - Allow automatic validation of Web pages
 - eliminate common coding errors
 - retain compatibility with “legacy” browsers
 - One of many XML languages
 - Big opportunity for e-commerce transactions and information exchange using XML vocabularies
 - Render document appropriately for different media and browsers

XML - pretty *HOT* stuff

➤ XHTML - transition between HTML and XML



HTML revisited
(why do I need XML anyhow?)

➤ HTML

- *Presentation* language
- Tags define how to display text
- HTML makes web stuff look “pretty”
- HTML describes *HOW TO DISPLAY*
- HTML does not *easily* describe page CONTENT

NOT meant for

- *Tight control* of document display
- *Flexibility* for different but specific types of info
- Rendering information in *variety of media and formats*
- *Publishing* single set of information *across this variety*
- Defining *complex linking* relationships

➤ HTML

- Lacks fine controls
 - Cannot specify display size of document
 - Cannot control window size of browser windows
- Display output varies depending on browser
 - Internet Explorer
 - Netscape Navigator
 - Mosaic
 - HotJava
 - Others

New tags and style sheets help with HTML

- IS EASY, QUICK, and CHEAP TO USE
 - HTML can be created with text editor and a little knowledge
- Is very forgiving of “errors”

XML - pretty *HOT* stuff

- HTML and XML
 - Both derive from SGML
 - Both use tags
 - Both use attributes
- HTML
 - says HOW text is displayed in browser window
- XML
 - says what each word means
 - Formal and precise
 - Flexible and growing
 - Structured
- XHTML
 - Reformulated HTML 4.0
 - Application of XML

- XML
 - Developed under WWW Consortium umbrella
 - That would be the W3C
 - Objective was to bring SGML (standard generalized markup language) to the web
 - SGML is a language designed to talk about other languages
 - Parent to HTML & XML (and many other markup languages)
 - SGML - the “kitchen sink” (and still under construction)
 - Has EVERYTHING anyone would EVER need for ANY markup language situation
 - SGML is a metalanguage
 - Extra bells and whistles because it needs to be EVERYTHING to EVERYONE
 - Makes implementation difficult and language slow
 - XML - the *trim and slim* implementation of SGML
 - Designed from the outset for the web

XML - pretty *HOT* stuff

- XML document
 - Well-formed (this is a basic XML rule)
 - Must have start & end tags for *every element*
 - One and *ONLY ONE* root element
 - Empty elements must be formatted correctly
 - *CASE* of start and end tags can be EITHER UPPERCASE or lowercase, but **MUST MATCH**
 - Elements must *NEST* correctly
 - Attribute values must *ALWAYS BE IN 'QUOTES'*
 - Single or double, just make them the SAME
 - Valid (*optional to validate*)
 - Contains **<!DOCTYPE declaration**
 - But unless it is checked it is not considered VALID
 - (Well-formed) AND VALIDATED against a DTD or SCHEMA
 - XML parser has determined document CONFORMS TO RULES
 - Can have Well-formed only or WELL-FORMED AND VALIDATED
 - There is an entire class of “non-validating processors” which *do not USE the DTD (Document Type Definition)*

➤ HTML vs XML - a simple example

```
<html>
<head>
<title>XML Sites</title>
</head>
<body>
<h1>Web sites on XML </h1>
<p>Here is a list of some <em><font color=red>XML Web sites</em></font>
<ul>
<li>http://www.xml.com
<li>http://www.xml.org
<li>http://alphaworks.ibm.com/xml
</ul>
<br>
>img src="file.gif" alt="image name">
</body>
</html>
```

This is the HTML code you are familiar with

XML - pretty *HOT* stuff

- HTML vs XML - a few changes make a correct XML example that looks VERY FAMILIAR! (after you add the ending tags and fix the order - required by XML)

```
<html>
<head>
<title>XML Sites</title>
</head>
<body>
<h1>Web sites on XML </h1>
<p>Here is a list of some <em><font color="red">XML Web sites</font></em></p>
<ul>
<li>http://www.xml.com</li>
<li>http://www.xml.org</li>
<li>http://alphaworks.ibm.com/xml</li>
</ul>
<br />

</body>
</html>
```

XML - pretty *HOT* stuff

➤ XML requirements

What I had to change from prior HTML code to conform to XML rules

- Elements are CASE SENSITIVE (must match)
- All elements MUST have *start* tag and *end* tag
 - Added the ending paragraph tag `</p>`
- Corrected the order of the closing tags for the `` (along with adding the `</p>`)
 - Must close in EXACT REVERSE ORDER
 - Close first what you opened last
- All attribute values must be in quotes (single or double just make them match)
 - Changed
 - `` to
 - ``
- Singleton or “empty” tags must be closed
 - Requires space followed by slash or `/tag`)
 - `
` becomes `
`
 - `` becomes ``

XML documents
the “parts”

XML - pretty *HOT* stuff

- **XML document - the parts**
 - XML declaration
 - Typically first item in document
 - Document Type Definition (DTD)
 - Required if document is to be validated via validating parser - which is what is also required to have a “valid” document
 - May be internal to document
 - May reference a URL
 - May be a combination of internal + URL
 - Document Element
 - Single element required
 - Root element or Document element
 - This is the top-level tag - you choose the name
 - All tags and contents live within this tag
 - Under HTML this was the <HTML> </HTML> tag combination
 - Document Content
 - Elements
 - Attributes
 - Comments
 - Processing Instructions

XML - pretty **HOT** stuff

➤ XML Declaration or prologue

- identifies the language
- Provides version number
 - Very similar to DOCTYPE element in HTML which looks like:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
```

- where in XML you would see:

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

- For XML the ? indicates a construct called a *processing instruction (or pi)*
- HTML typically uses uppercase (but you can write it anyway you want Html HTml html, etc since folding occurs)
- XML cares about case (this also makes XML global)
- Simplest doctype declaration is one that references URL

XML - pretty *HOT* stuff

➤ XML Document Element

- Must have single element containing all other elements
- This is the *ROOT ELEMENT* or *DOCUMENT ELEMENT*
 - Note that in HTML this is <HTML> since all other content is inside the begin/end tags <HTML> </HTML>
- In XML you CANNOT use XML as root (XML is reserved)
 - Here is an example where *MEMO* is the document element!

```
<?xml version="1.0"?.  
<!DOCTYPE MEMO "http://www.jmw.com/memo.dtd"  
<MEMO>  
<TO>William Winchell</TO>  
<CC>Donald McCord</CC>  
<FROM>Janice Winchell</FROM>  
<DATE>10/11/2003</DATE>  
<RE>Upcoming travel plans</RE>  
<BODY>The next planned trip will be in November to the  
IBM Technical Conference in Las Vegas from November 10-12, 2003.  
Please have all your travel documents ready, and your presentation  
material ready to ship by October 3, 2003. </BODY>  
</MEMO>
```

➤ Document Content

➤ Elements

- Primary components of a document
- Bound by start and end tags
- May have CDATA(character not markup data) describing element
- May be “empty”
 - That is *no content* between open and close tag
 - *May* have attributes

 is just as empty as

 or
</br>

➤ Attributes

- Property of an element
- information *WITHIN* an element’s start tag
- <car type = “MGB”></car>
- Composed as *name=value* pair
 - *name* is name of attribute (type)
 - *value* is the value(MGB)

XML - pretty *HOT* stuff

➤ XML Comments

Syntactically equivalent to HTML and SGML comment

- Begin with markup declaration open delimiter `<!`
 - and then a comment open delimiter `--`
 - So, you end up with `<!--`
 - Close comment with a close delimiter and a markup declaration close delimiter, or `-->`
 - Any character data may be contained in the comment
 - *Not the > character*
 - *Not the double hyphens --*
 - Comments are important for later reading (just as in any programming language)
 - Comments may appear anywhere
 - Must be outside other markup
 - Not immediately prior to eh XML declaration
- `<?xml version="1.0"?>`**
- Here is a valid comment:
`<!-- Put your comment here -->`
 - Note that SGML allows “inline comments” - XML does NOT

XML - pretty *HOT* stuff

➤ **Processing Instructions**

- Provide means to send instructions to
 - Computer Program
 - Application
- Bounded by `<? and ?>`
- The information is passed through to the application
- An XML parser *may or may not pass on comments*
- A conforming parser **MUST** pass on processing instructions

`<?noisemaker noise='cello.wav' ?>`

- The application is noisemaker
 - The attribute noise and its corresponding value cello.wav
 - The noisemaker application should now make the noise contained within the cello.wav sound file
- Another typical example involves style sheets in an XML doc

`<?xml:stylesheet type="text/xsl"?>`

- or how about a familiar one for javascript:

`<?javascript (javascript stuff) ?>` or

`<?Tcl (tcl stuff) ?>`

XML

the grammar rules

- **XML employs the Extended Backus-Naur Form (EBNF) (with added commentary)**

Note: John Backus and Peter Naur described formal notation for describing syntax of ALGOL60 computer language - still a pretty good model today and what the W3C group used for XML (plus comments)

- Basic notation:
 - symbol ::= expression
 - where ::= represents “is defined as”
 - Example: new symbol vowels ::= [aeiou]
 - Rules can refer to range of sequential characters
 - Example: FirstFive ::= [a-e] representing the first 5 characters of the English alphabet
 - Remember that **case counts** so maybe we should say
 - vowel ::= [aeiouAEIOU]if we want both upper and lower case to apply

➤ **EBNF**

Can also handle complex expressions

such as “one or more of these options:

- $S ::= (\#x20 \mid \#x9 \mid \#xD \mid \#xA) +$
 - S (represents white space)
 - $\#x20$ is hex 20 (ASCII 32)
 - $\#x9$ is the tab
 - $\#xD$ is the newline
 - $\#xA$ is the linefeed
 - The \mid is representative of an “or”
 - The $+$ is “one or more”

So this rule reads:

White space is defined as one or more of the space, tab, newline, or linefeed characters

➤ **EBNF**

Can also do One But Not The Other and Either/Or

- AcceptableGrade ::= A - B becomes the rule
 - An acceptable grade is defined as the character A but NOT the character B
- AcceptableGrade ::= A | B
 - An acceptable grade can be EITHER A OR B

Excluding Characters from a set:

- consonants ::= ([^aeiou] | alphabet)
 - Consonants are defined as the alphabet, except the letters a, e, i, o, u
 - The pipe character, |, separates the subject of the expression (alphabet) from the exclusion set
- XML grammar rules are written in EBNF, but as you can see this language is fairly TERSE

Document Type Definition

DTD

- **Document Type Definition (DTD)**
 - Sets forth the rules for what makes the document VALID
 - Can be written “internally” right inside doctype definition
 - Can be referenced via URL
 - Might have combination of the 2!
 - Similar to style sheet in XML
 - Identifies the grammar expected
 - Identifies the elements
 - Tells how to use element
 - Builds the structure and nesting
 - Identifies the hierarchy

Good practice to declare everything before using
Makes it easier to find/dissect the information

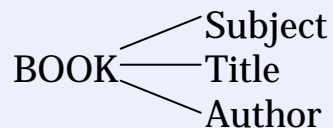
Rules of the DTD road

TERM	Example	What it does
XML declaration	<code><?xml version = "1.0" ?></code>	Identifies version of XML to use
Document type declaration	<code><!DOCTYPE Root-Element SYSTEM "Root-Element.dtd"></code>	Identifies where to find the dtd (presuming it is external but in the same directory (URI))
Element type declaration	<code><!ELEMENT Name (#PCDATA)></code>	Defines element type
Attribute list declaration	<code><!ATTLIST Element-Name Name Data-Type Default></code>	Defines name, data type and a default value for each attribute
Entity declaration	<code><!ENTITY Entity-Name "text"></code>	Defines information that can be "called" using entity name
Notation declaration	<code><!NOTATION Name System 'externalID"></code>	Association name to find the interpreter for the notation

XML - pretty *HOT* stuff

- ▶ **DTD Example (inline)**
 - ▶ Comments are always appreciated
 - ▶ Never nest comment in another tag
 - ▶ Never use a hyphen in your comment
 - ▶ ELEMENT defines the name (and more)
 - ▶ ENTITY defines alias for block of text or external content

```
<!DOCTYPE Book-Review [  
<!ELEMENT Book-Review (Book*)>  
<!-- Only a single subject, single title, and single author are expected/allowed -->  
<!--Data is parsed character data (#PCDATA) (same as text) -->  
<!--Book has 3 subordinate items, Subject, Title, and Author) -->  
<!ELEMENT Book (Subject, Title, Author)>  
<!ELEMENT Subject (#PCDATA)>  
<!ELEMENT Title (#PCDATA)>  
<!ELEMENT Author (#PCDATA).  
<!ENTITY Publisher "Winchell Books">  
>
```



XML - pretty *HOT* stuff

➤ **DTD**

- Book-Review is the root element
- Element type declarations name the elements and identify children
 - <!ELEMENT Name ANY> can have data or markup info
 - <!ELEMENT Name EMPTY> must contain no content
 - <!ELEMENT Name (#PCDATA(| ChildName))>
 - <!ELEMENT Name (Child1, Child2)>

```
<?xml version = '1.0' encoding='UTF-8' standalone='yes' ?>
<!DOCTYPE Book [
<!ELEMENT Book-Review (Book*)>
<!-- Subject is optional or repeatable, multiple authors allowed -->
<!-- Order is as specified, subject then title then author -->
<!ELEMENT Book (Subject*, Title, Author*)>
<!ELEMENT Subject (#PCDATA)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Author (#PCDATA)>
<!ENTITY Publisher "Winchell Books">
]>
```

XML - pretty *HOT* stuff

- **DTD**

- **Entities**

- Provide “shorthand”
- May be a placeholder
- Works like parameter substitution
- Two “high-level” types
 - General
 - Parameter
- **Four subsets**
 - **Internal entities**
 - References to definitions entirely within DTD
 - **External**
 - References to definitions OUTSIDE the document DTD
 - **Parsed**
 - XML processor can and will parse
 - **Unparsed**
 - Handed off to another application

XML - pretty *HOT* stuff

➤ **General Entities**

- Created in DTD
- Used in XML document

➤ **Parameter Entities**

- Created AND USED in the DTD

➤ **Predefined or “built-in” Entities (no need to declare these)**

< <

> >

' ‘

"e; “

& &

➤ **Benefits of using entities beyond just parameterization**

- Create reusable - can create separate pieces and use external entities to “include” them

XML - pretty *HOT* stuff

➤ **DTD how to get started**

- **Step 1 is to analyze your data!!!**
 - **Create data structure (data and children)**
- Step 2 - Define element names (this is your data dictionary)
- Consider putting information into parameter ENTITY
 - Allows easier changing later and makes DTD global ready
- ENTITY makes it easier to translate entity declarations without changing the underlying DTD
 - Publisher is the English word
 - Verleger is the German word for publisher

```
<!-- Element names -->
<!ENTITY % Doctype          "doctype"
<!ENTITY % Title            "title">
<!ENTITY % Author           "author">
<!ENTITY % Date             "date">
<!ENTITY % Publisher        "publisher">
<!--Document root -->
<!ELEMENT %Publisher;
```

XML - pretty *HOT* stuff

➤ ENTITY markup rules

- Must be declared in DTD or schema
 - If document not being validated must create enough of a DTD within your XML to at least declare the entity you want to use (excepting the pre-defined ones)
- *General entity* reference must have &myEntity; format when used in doc
- Name must *begin* with letter or underscore
 - Can contain letters, underscores, whole numbers, colons, periods and/or dashes
- Declaration cannot have markup the begins in entity declaration and ends outside
- *Parameter entity* must have preceding percent sign (%) with white space before and after, and then reference with %name no space

```
<!-- Element names -->
<!ENTITY % Doctype          "doctype"
<!ENTITY % Title            "title">
<!ENTITY % Author          "author">
<!ENTITY % Date            "date">
<!ENTITY % Publisher        "publisher">
<!--Document root -->
...
<!ELEMENT %Publisher;
```

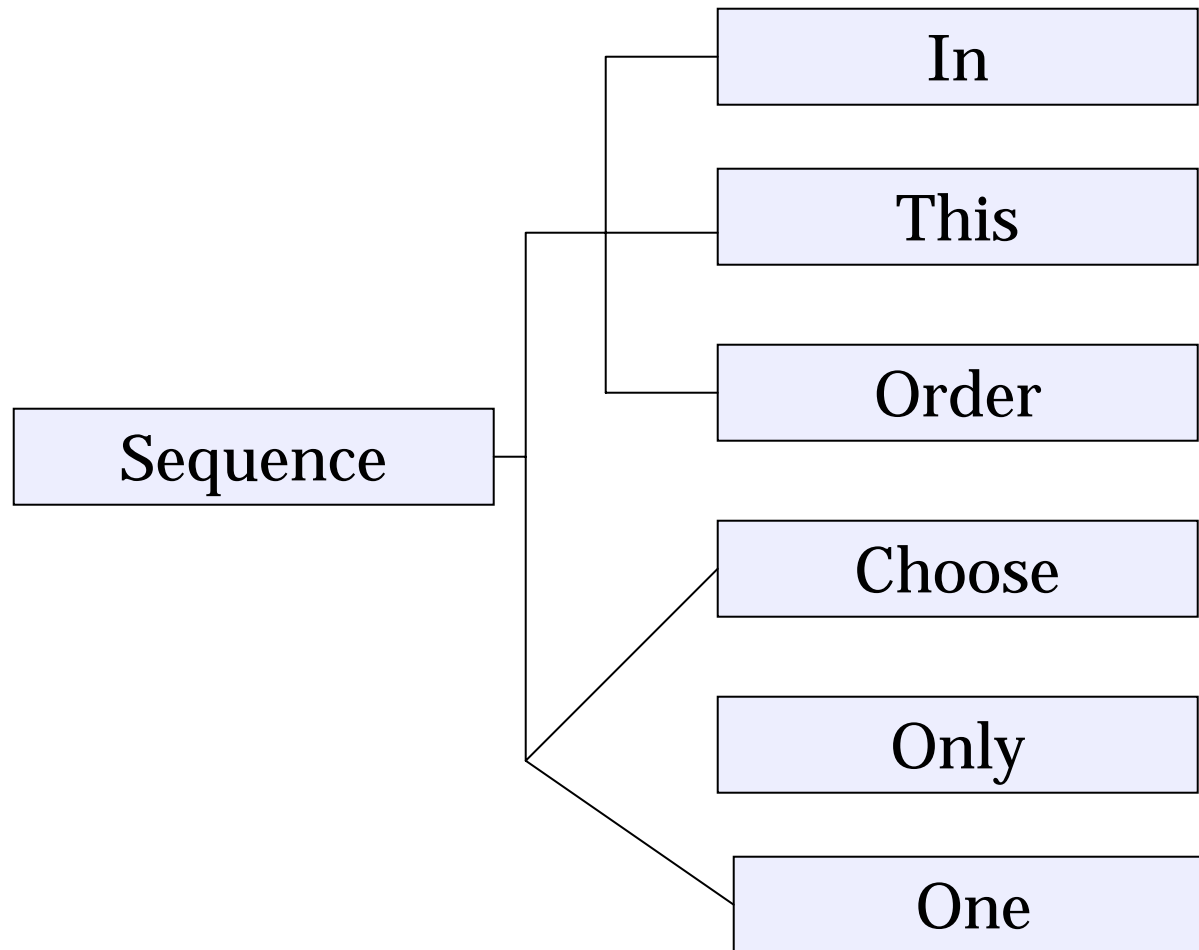
XML - pretty *HOT* stuff

➤ XML Entity coding sample snippet from XML site

```
<?xml version='1.0' encoding='UTF-8'?>
<?xml-stylesheet type="text/xsl" href="xmlspec.xsl"?>
<!DOCTYPE spec PUBLIC "-//W3C//DTD Specification V2.1//EN" "xmlspec-v21.dtd" [
<!ENTITY year "2001">
<!ENTITY month "June">
<!ENTITY MM "06">
<!ENTITY day "27">
<!ENTITY DD "27">
<!ENTITY internalXMLBase "http://www.w3.org/XML/Group/&year;/&MM;/REC-xmlbase-&year;&MM;&DD;">
<!ENTITY externalXMLBase "http://www.w3.org/TR/&year;/REC-xmlbase-&year;&MM;&DD;">
<!ENTITY XMLBase "&externalXMLBase;">
<!ENTITY LatestXMLBase "http://www.w3.org/TR/xmlbase">
]>
<spec w3c-doctype="rec" status="final">
<header>
<title>XML Base</title>
<w3c-designation>xml-base-&year;&MM;&DD;</w3c-designation>
<w3c-doctype>W3C Recommendation</w3c-doctype>
<pubdate><day>&day;</day><month>&month;</month><year>&year;</year></pubdate>
<publoc>
<loc href="&XMLBase;/">&XMLBase;</loc> (available in
<loc role="available-format" href="&XMLBase;/Overview.html">HTML</loc>,
<loc role="available-format" href="&XMLBase;/Overview.xml">XML</loc>)
</publoc>
<prevlocs><loc href="http://www.w3.org/TR/2000/PR-xmlbase-20001220/">http://www.w3.org/TR/2000/PR-xmlbase-20001220
</loc>
<!-- <loc href="http://www.w3.org/TR/2000/CR-xmlbase-20000908">http://www.w3.org/TR/2000/CR-xmlbase-20000908</loc>
<loc href="http://www.w3.org/TR/2000/WD-xmlbase-20000607">http://www.w3.org/TR/2000/WD-xmlbase-20000607</loc>
<loc href="http://www.w3.org/TR/2000/WD-xmlbase-20000221">http://www.w3.org/TR/2000/WD-xmlbase-20000221</loc>
<loc href="http://www.w3.org/TR/1999/WD-xmlbase-19991220">http://www.w3.org/TR/1999/WD-xmlbase-19991220</loc>-->
</prevlocs>
<latestloc>
<loc href="&LatestXMLBase;/">&LatestXMLBase;</loc>
</latestloc>.....
```

XML - pretty *HOT* stuff

- **Structure charts help define data and hierarchy**



This example is representative of structure charts used for XML design by Microstar Software Ltd.

XML - pretty *HOT* stuff

- **Check whether your industry already has a DTD- here are a few that already exist (keep checking)**

Reference DTD

Use or industry

Reference DTD	Use or industry
TCIF	Telecommunication Industry Forum Standard
SIF	Schools Interoperability Framework
HL7	Health industry for charting and diagnosis
TEI	Text Encoding Initiative - book oriented
HTML	Web publishing
ISO 12083	Electronic Manuscript Standard for journals, articles, etc
SAE J2008	Spec for documentation of Vehicle Service information
DocBook	Specs for Software Documentation
ATA Spec 2100	Airframe/Engine documentation spec
MIL38784B (CALs)	Military Specification Technical Documentation

Namespaces

➤ **Why do we need namespaces?**

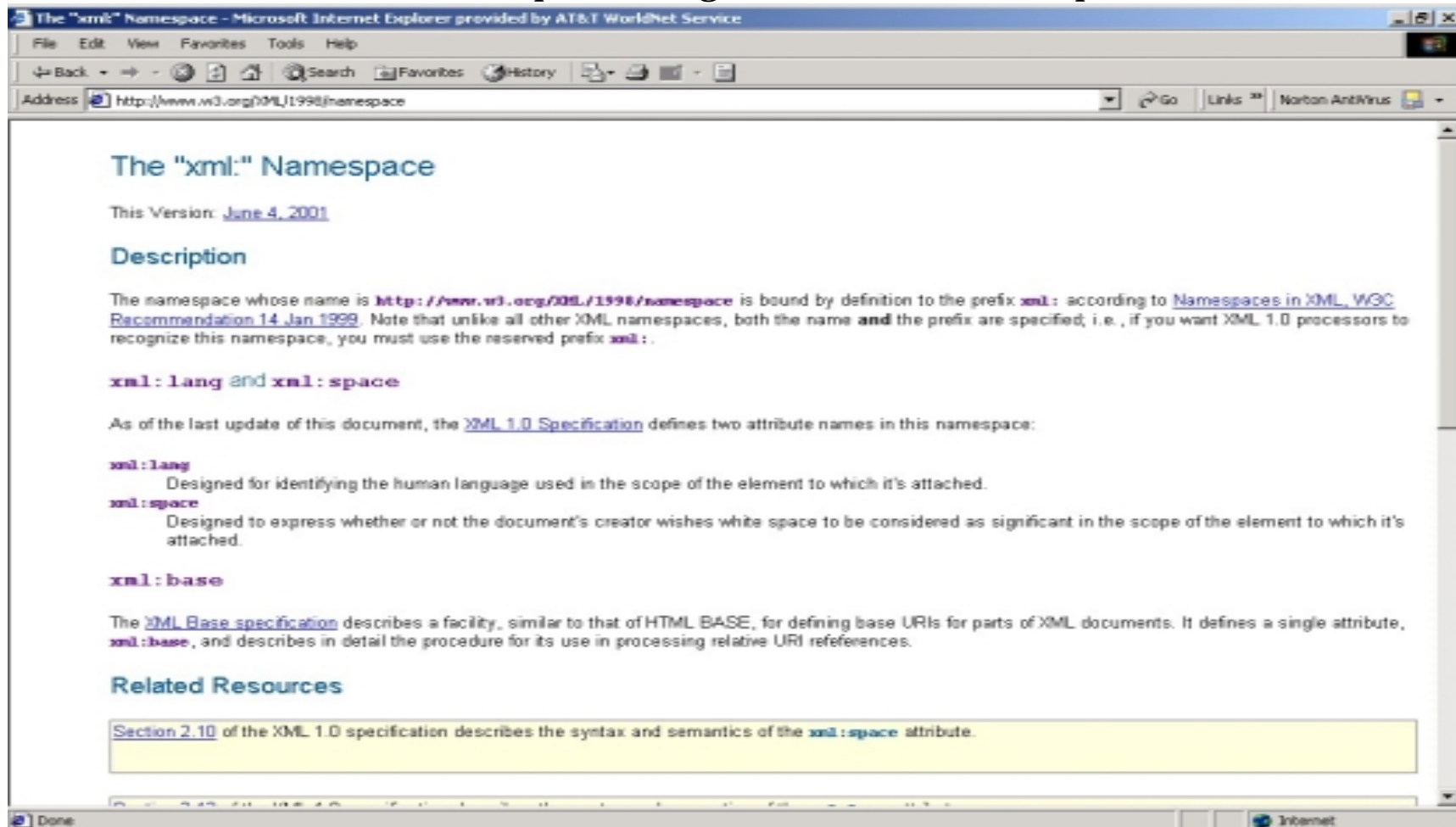
- With flexibility comes conflict
 - Documents can be created using the SAME NAMES
 - RECORD might mean a music element OR it might mean a person's vehicle data, or a medical piece
- Namespaces allow collision avoidance
 - Can use multiple DTD's in the same document and ensure unique name
 - Within a DTD a name MUST be unique
- Technical definition of namespace from the XML

Recommendation:

“An XML *Namespace* is a collection of names, identified by a URI reference, which are used in XML documents as element types and attributes”

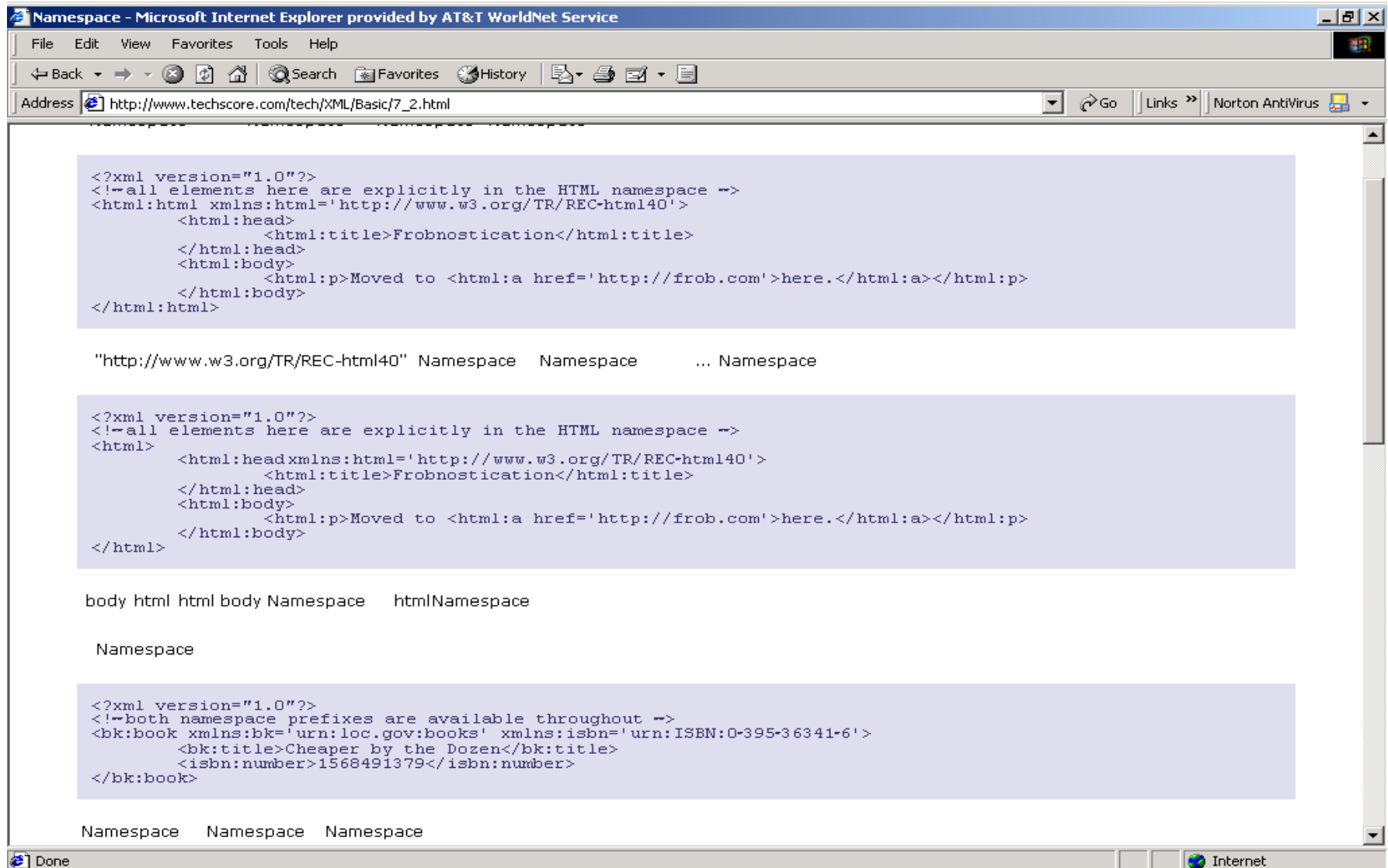
XML - pretty *HOT* stuff

- ▶ **Some namespaces already exist**
 - ▶ XML itself:
 - ▶ `xmlns:xml="http://w3.org/XML/1998/namespace"`



XML - pretty *HOT* stuff

▶ Here are some namespace coding examples



The screenshot shows a Microsoft Internet Explorer browser window with the address bar displaying `http://www.techscore.com/tech/XML/Basic/7_2.html`. The browser's status bar at the bottom indicates "Done" and "Internet".

The main content area displays three XML code snippets, each followed by a list of namespace declarations:

```
<?xml version="1.0"?>
<!--all elements here are explicitly in the HTML namespace -->
<html:html xmlns:html='http://www.w3.org/TR/REC-html40'>
  <html:head>
    <html:title>Frobnostication</html:title>
  </html:head>
  <html:body>
    <html:p>Moved to <html:a href='http://frob.com'>here.</html:a></html:p>
  </html:body>
</html:html>
```

"http://www.w3.org/TR/REC-html40" Namespace Namespace ... Namespace

```
<?xml version="1.0"?>
<!--all elements here are explicitly in the HTML namespace -->
<html>
  <html:headxmlns:html='http://www.w3.org/TR/REC-html40'>
    <html:title>Frobnostication</html:title>
  </html:head>
  <html:body>
    <html:p>Moved to <html:a href='http://frob.com'>here.</html:a></html:p>
  </html:body>
</html>
```

body html html body Namespace htmlNamespace

Namespace

```
<?xml version="1.0"?>
<!--both namespace prefixes are available throughout -->
<bk:book xmlns:bk='urn:loc.gov:books' xmlns:isbn='urn:ISBN:0-395-36341-6'>
  <bk:title>Cheaper by the Dozen</bk:title>
  <isbn:number>1568491379</isbn:number>
</bk:book>
```

Namespace Namespace Namespace

➤ **Declaring a namespace**

- Use XML attribute xmlns
- Available in EVERY xml element even if not declared in a DTD
- xmlns is part of the XML language
- Create namespace by including xmlns in start tag or root element of document
- Scope of namespace is that single element and all its children
- XHTML has a namespace
 - Designed to be easily extended using other XML DTD's
 - *Requires namespace declaration in every XHTML doc to be considered valid*

XML - pretty *HOT* stuff

- ▶ **XHTML requires namespace**
 - ▶ Must declare in every XHTML document
 - ▶ Namespace for XHTML 1.0 is `http://222.23.org/1999/xhtml`
 - ▶ If no other namespaces are used or if it is the only namespace you can declare XHTML as the “default” namespace and you won’t need to prefix tags just declare it:

```
<html xmlns="http://www.w3.org/1999/xhtml">  
...  
</html>
```

- ▶ Because there is no `:name=` in the `xmlns` declaration this becomes the “default” namespace
- ▶ Consider which tags will be most used and make that your default or primary namespace

```
<html xmlns="http://www.w3.org/1999/xhtml"  
      xmlns:htm="http://www.w3.org/TR/REC-html140">  
<htm:p>Some text in the HTML 4.0 namespace</htm:p>  
<p>Some text in the XHTML namespace</p>  
...  
</html>
```

- ▶ **Namespace “Issues”**
 - ▶ Seriously deficient for internationalization
 - ▶ URI syntax depends on US-ASCII
 - ▶ If alphabet not based upon Roman alphabet then what?
 - ▶ Some accommodations already used:
 - ▶ ue for German ü as an example
 - ▶ Many languages cannot be handled
 - ▶ Cyrillic, Arabic, Japanese, Chinese, Hebrew, etc....
 - ▶ Proposals are out to extend URI to include support for UTF-8 ISO-10646 character sets
 - ▶ Problem is, now we look like strange gobldygook language:
 - ▶ `http://www.أليس:.om/`
 - ▶ Overriding names using namespace prefix means you get to add prefix names to your style sheet in order to use it
 - ▶ Namespaces are considered a temporary solution waiting on XML Schemas

XML Schemas

- ▶ **Schemas are the „Next“ generation**
 - ▶ Plan is to replace DTD
 - ▶ Allow boundary and error checking
 - ▶ Structured records can be checked for being well-formed and for being valid
 - ▶ Address issue whereby XML allows information of ANY length to be in a CDATA field, database might accept only 20
 - ▶ Want to be able to specify length and nature of information
 - ▶ Eliminate having to traffic over the internet to find these kinds of “mistakes”
 - ▶ Base for starting to find info about SCHEMA would be:

<http://www.w3.org/TR/NOTE-xml-schema-req>

➤ **Schema - model for the content**

➤ **Also allows constraints**

- Where element may be used
- When element may be used
- Datatype constraints
 - Data size
 - Data content

➤ Consider an XML DTD with zip-code

```
<!ELEMENT zip-code #PCDATA>
```

- sets up for interesting data for the zip-code

```
<zip-code> JMW-1987345888888888-xyz</zip-code >
```

- The content is well-formed and “valid” - it is Parsed Character Data - but it is certainly not a valid ZIP CODE
- Datatype constraints in schemas can limit content to 5-digit number or 9-digit number broken between 5 and 6 by a hyphen
 - Certainly would improve data reliability with less overhead!

- **DTD and SCHEMA compared:**
 - Language used
 - DTD - EBNF
 - Schema - XML itself
 - Data constraint options
 - DTD - minimal - CDATA but not much more
 - Attributes can help with constraints using enumerated values
 - Not practical where large number of value choices
 - Schema - Specific constraints can be specified
 - User-Defined Types
 - DTD - limited to fixed set of content model
 - Schema - allows for flexibility in expressing content as well as in limiting content

XML - pretty *HOT* stuff

➤ **Information from:** <http://www.w3.org/TR/xmlschema-0/>

XML Schema Part 0: Primer - Microsoft Internet Explorer provided by AT&T WorldNet Service

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History

Address <http://www.w3.org/TR/xmlschema-0/> Go Links Norton AntiVirus

<http://www.w3.org/TR/xmlschema-0/>

A list of current W3C Recommendations and other technical documents can be found at <http://www.w3.org/TR>.

Table of contents

W3C Recommendation

- 1 [Introduction](#)
- 2 [Basic Concepts: The Purchase Order](#)
 - 2.1 [The Purchase Order Schema](#)
 - 2.2 [Complex Type Definitions, Element & Attribute Declarations](#)
 - 2.2.1 [Occurrence Constraints](#)
 - 2.2.2 [Global Elements & Attributes](#)
 - 2.2.3 [Naming Conflicts](#)
 - 2.3 [Simple Types](#)
 - 2.3.1 [List Types](#)
 - 2.3.2 [Union Types](#)
 - 2.4 [Anonymous Type Definitions](#)
 - 2.5 [Element Content](#)
 - 2.5.1 [Complex Types from Simple Types](#)
 - 2.5.2 [Mixed Content](#)
 - 2.5.3 [Empty Content](#)
 - 2.5.4 [anyType](#)
 - 2.6 [Annotations](#)
 - 2.7 [Building Content Models](#)
 - 2.8 [Attribute Groups](#)
 - 2.9 [Nil Values](#)
- 3 [Advanced Concepts I: Namespaces, Schemas & Qualification](#)
 - 3.1 [Target Namespaces & Unqualified Locals](#)
 - 3.2 [Qualified Locals](#)
 - 3.3 [Global vs. Local Declarations](#)
 - 3.4 [Undeclared Target Namespaces](#)
- 4 [Advanced Concepts II: The International Purchase Order](#)
 - 4.1 [A Schema in Multiple Documents](#)
 - 4.2 [Deriving Types by Extension](#)
 - 4.3 [Using Derived Types in Instance Documents](#)
 - 4.4 [Deriving Complex Types by Restriction](#)
 - 4.5 [Redefining Types & Groups](#)

Internet

XML - pretty *HOT* stuff

Information from:

<http://www.w3.org/TR/xmlschema-0/#Intro/>

1 Introduction

This document, XML Schema Part 0: Primer, provides an easily approachable description of the XML Schema definition language, and should be used alongside the formal descriptions of the language contained in Parts [1](#) and [2](#) of the XML Schema specification. The intended audience of this document includes application developers whose programs read and write schema documents, and schema authors who need to know about the features of the language, especially features that provide functionality above and beyond what is provided by DTDs. The text assumes that you have a basic understanding of [XML 1.0](#) and [XML-Namespaces](#). Each major section of the primer introduces new features of the language, and describes those features in the context of concrete examples.

[Section 2](#) covers the basic mechanisms of XML Schema. It describes how to declare the elements and attributes that appear in XML documents, the distinctions between simple and complex types, defining complex types, the use of simple types for element and attribute values, schema annotation, a simple mechanism for re-using element and attribute definitions, and nil values.

[Section 3](#), the first advanced section in the primer, explains the basics of how namespaces are used in XML and schema documents. This section is important for understanding many of the topics that appear in the other advanced sections.

[Section 4](#), the second advanced section in the primer, describes mechanisms for deriving types from existing types, and for controlling these derivations. The section also describes mechanisms for merging together fragments of a schema from multiple sources, and for element substitution.

[Section 5](#) covers more advanced features, including a mechanism for specifying uniqueness among attributes and elements, a mechanism for using types across namespaces, a mechanism for extending types based on namespaces, and a description of how documents are checked for conformance.

In addition to the sections just described, the primer contains a number of [appendices](#) that provide detailed reference information on simple types and a regular expression language.

The primer is a non-normative document, which means that it does not provide a definitive (from the W3C's point of view) specification of the XML Schema language. The examples and other explanatory material in this document are provided to help you understand XML Schema, but they may not always provide definitive answers. In such cases, you will need to refer to the XML Schema specification, and to help you do this, we provide many links pointing to the relevant parts of the specification. More specifically, XML Schema items mentioned in the primer text are linked to an [index](#) of element names and attributes, and a summary [table](#) of datatypes, both in the primer. The table and the index contain links to the relevant sections of XML Schema parts 1 and 2.

2 Basic Concepts: The Purchase Order

The purpose of a schema is to define a class of XML documents, and so the term "instance document" is often used to describe an XML document that conforms to a particular schema. In fact, neither instances nor schemas need to exist as documents *per se* -- they may exist as streams of bytes sent

- ▶ **Working group on SCHEMA describes requirements:**
 1. Provide for primitive datatyping, including byte, date, integer, sequence, SQL and Java primitive datatypes, etc
 2. Define type system that is adequate for import/export from database systems
 3. Distinguish requirements of lexical data representation vs. requirements controlling underlying information set
 4. Allow creation of user-defined datatypes
 1. May be derived from existing and which may constrain properties (range, precision, length, mask, etc)

XML - pretty *HOT* stuff

- **Primitive datatypes built in to XML schema (Base of these 14 datatypes)**

ID	NOTATION
IDREF	string
IDREFS	boolean
ENTITY	number
ENTITIES	dateTime
NMTOKEN	binary
NMTOKENS	uri

- **Plus these added types derived from the original 14**

integer	decimal
real	time
timePeriod	

- **User-Generated datatypes**
 - Takes an existing datatype
 - Adds constraints
 - Number is a base type
 - A number which must be positive imposes a constraint
 - Decimal is a form of integer
 - A number which must be between 0.0 and 6.0 constrains value to a minimum and maximum value
 - Precision and scale can impose constraint
 - Max value allowed constraints...
 - well....you get the idea

- ▶ **XML Schema is still a working draft**
 - ▶ Use schemas with care, as standards may change
 - ▶ Schemas will “probably” replace DTD
 - ▶ More flexibility
 - ▶ More capability
 - ▶ Here is a sample where the “phone” datatype allows either 10 (with area code) or 7 (no area code) digits

```
<datatype name="phone">
  <basetype name="string"/>
  <lexicalRepresentation>
    <lexical>999-999-9999</lexical>
    <lexical>999-9999</lexical>
  </lexicalRepresentation>
</datatype>
```

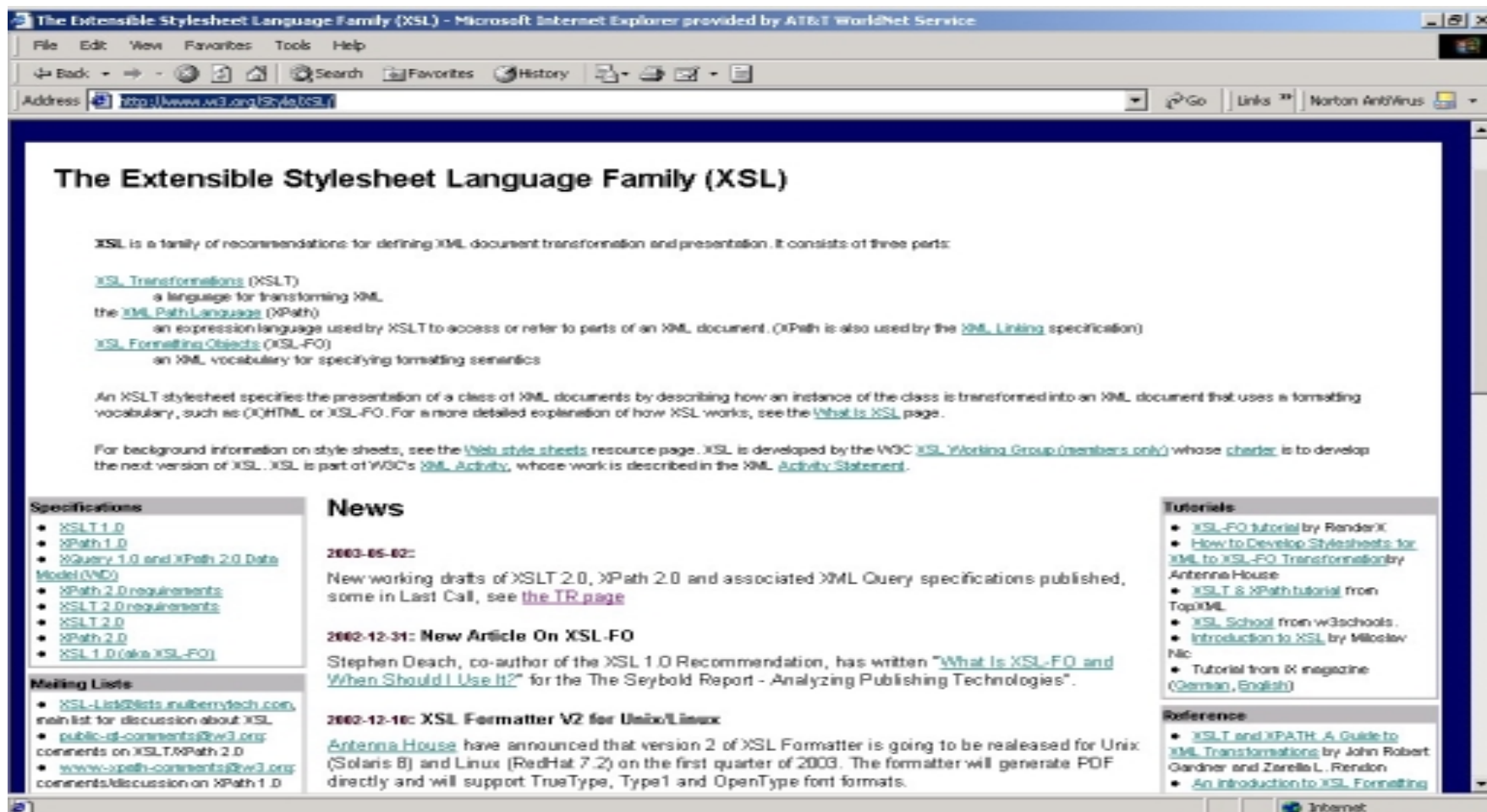
XSL

eXtensible Style sheet Language

XML - pretty *HOT* stuff

➤ Current information on XSL look to:

<http://www.w3.org/Style/XSL/>



The screenshot shows a Microsoft Internet Explorer browser window displaying the W3C website for the Extensible Stylesheet Language Family (XSL). The browser's address bar shows the URL <http://www.w3.org/Style/XSL/>. The page title is "The Extensible Stylesheet Language Family (XSL)".

The main content of the page includes:

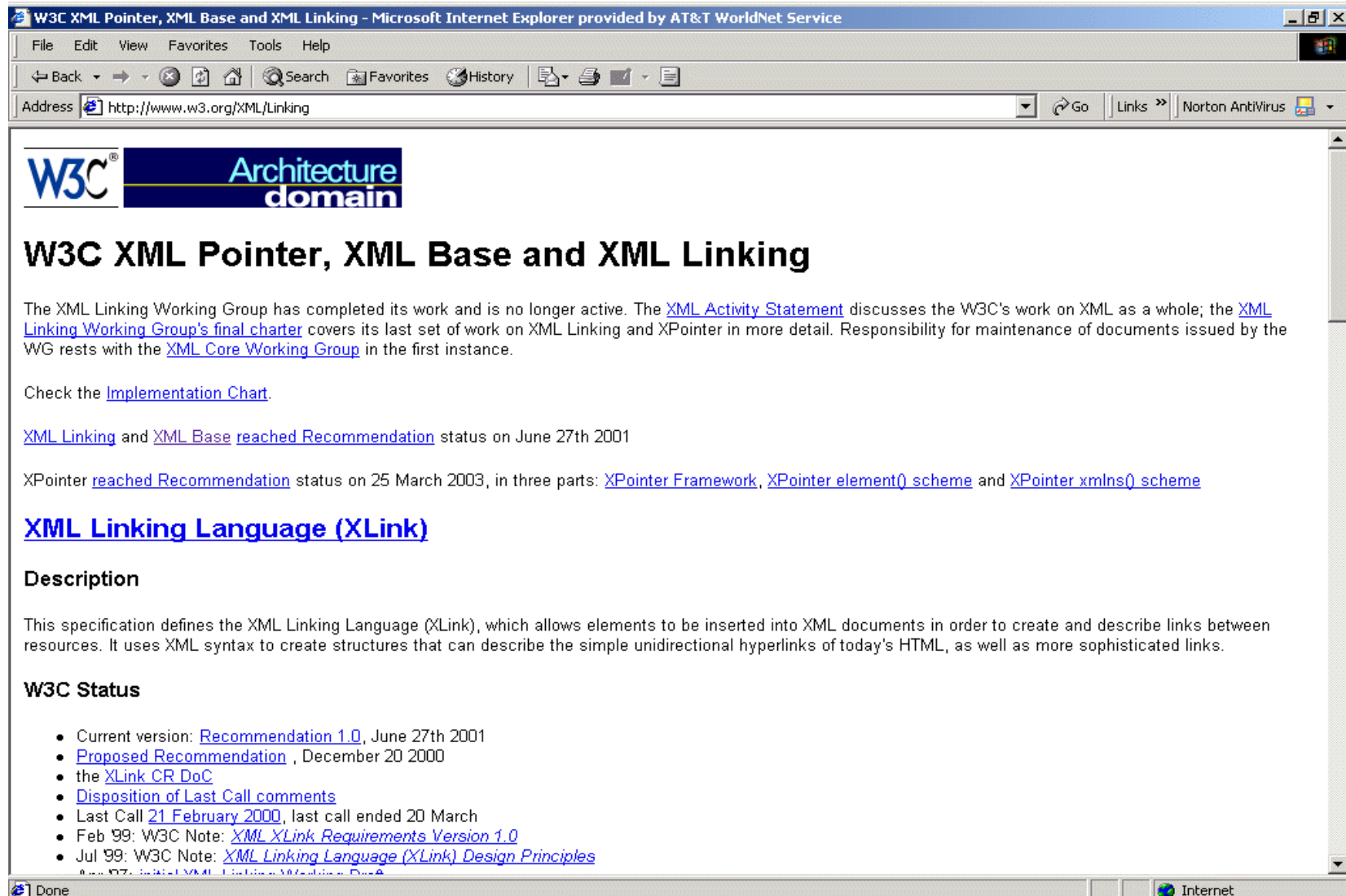
- Introduction:** "XSL is a family of recommendations for defining XML document transformation and presentation. It consists of three parts:"
- Specifications:**
 - [XSL Transformations \(XSLT\)](#): a language for transforming XML.
 - [XML Path Language \(XPath\)](#): an expression language used by XSLT to access or refer to parts of an XML document. (XPath is also used by the [XML Linking](#) specification)
 - [XSL Formatting Objects \(XSL-FO\)](#): an XML vocabulary for specifying formatting semantics.
- News:**
 - 2003-05-02:** New working drafts of XSLT 2.0, XPath 2.0 and associated XML Query specifications published, some in Last Call, see [the TR page](#)
 - 2002-12-31: New Article On XSL-FO**
Stephen Deach, co-author of the XSL 1.0 Recommendation, has written "[What Is XSL-FO and When Should I Use It?](#)" for the The Seybold Report - Analyzing Publishing Technologies".
 - 2002-12-10: XSL Formatter V2 for Unix/Linux**
[Antenna House](#) have announced that version 2 of XSL Formatter is going to be released for Unix (Solaris 8) and Linux (RedHat 7.2) on the first quarter of 2003. The formatter will generate PDF directly and will support TrueType, Type1 and OpenType font formats.
- Tutorials:**
 - [XSL-FO tutorial](#) by RenderX
 - [How to Develop Stylesheets for XML to XSL-FO Transformations](#) by Antenna House
 - [XSLT & XPath tutorial](#) from TopXML
 - [XSL School](#) from w3schools.
 - [Introduction to XSL](#) by Miroslav Nic
 - Tutorial from IX magazine (German, English)
- Reference:**
 - [XSLT and XPATH: A Guide to XML Transformations](#) by John Robert Gardner and Zarella L. Rendon
 - [An Introduction to XSL Formatting](#)

The browser interface includes a menu bar (File, Edit, View, Favorites, Tools, Help), a toolbar with navigation buttons, and a status bar at the bottom showing "Internet".

- ▶ **XSL**
 - ▶ Really 3 separate languages in various development stages
 - ▶ XML vocabulary can describe the formatting objects portion of XSL
 - ▶ XSLT or XSL transformations
 - ▶ markup vocabulary describing the TRANSFORMATION portion of XSL
 - ▶ XML Path language (XPath)
 - ▶ Language (not technically XML) used to address document fragments (portions of XML document)
 - ▶ This language also used by XSLT (also XPointer and XLink)
 - ▶ Transformations can be applied to document fragments
 - ▶ XSL searches for *pattern* and series of *one or more templates* that match the pattern
 - ▶ Returns a *result tree*
 - ▶ Allows for working with a small portion of an XML document

XML - pretty *HOT* stuff

➤ XML Pointer and Linking



The screenshot shows a Microsoft Internet Explorer browser window with the address bar displaying <http://www.w3.org/XML/Linking>. The page content includes the W3C logo and the text "Architecture domain". The main heading is "W3C XML Pointer, XML Base and XML Linking". The text below the heading states: "The XML Linking Working Group has completed its work and is no longer active. The [XML Activity Statement](#) discusses the W3C's work on XML as a whole; the [XML Linking Working Group's final charter](#) covers its last set of work on XML Linking and XPointer in more detail. Responsibility for maintenance of documents issued by the WG rests with the [XML Core Working Group](#) in the first instance." Below this, there are several links: "Check the [Implementation Chart](#).", "[XML Linking](#) and [XML Base](#) [reached Recommendation](#) status on June 27th 2001", and "XPointer [reached Recommendation](#) status on 25 March 2003, in three parts: [XPointer Framework](#), [XPointer element\(\) scheme](#) and [XPointer xmlns\(\) scheme](#)". A section titled "XML Linking Language (XLink)" follows, with a sub-section "Description" stating: "This specification defines the XML Linking Language (XLink), which allows elements to be inserted into XML documents in order to create and describe links between resources. It uses XML syntax to create structures that can describe the simple unidirectional hyperlinks of today's HTML, as well as more sophisticated links." Below that is a "W3C Status" section with a bulleted list: "• Current version: [Recommendation 1.0](#), June 27th 2001", "• [Proposed Recommendation](#), December 20 2000", "• the [XLink CR DoC](#)", "• [Disposition of Last Call comments](#)", "• Last Call [21 February 2000](#), last call ended 20 March", "• Feb 99: W3C Note: [XML XLink Requirements Version 1.0](#)", "• Jul 99: W3C Note: [XML Linking Language \(XLink\) Design Principles](#)". The browser's status bar at the bottom shows "Done" and "Internet".

Additional resources
where can you go from here?

XML - pretty *HOT* stuff

- **Obviously a LOT of information is available and much has been listed throughout this presentation**
- **Most of the information is available on the internet**

<http://www.w3.org/>

This would be “home”

<http://www.alphaworks.ibm.com/xml/>

alphaworks xml location

<http://www-106.ibm.com/developerworks/xml/>

developerworks xml site

<http://www.w3schools.com/>

free tutorials available for most all of the
XML technologies and pieces

<http://www.xml.com>

has more tutorials as well as live web-casts

XML - pretty *HOT* stuff

▶ Check out alphaWorks.ibm.com

alphaWorks : XML - Microsoft Internet Explorer provided by AT&T WorldNet Service

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Mail

Address http://www.alphaworks.ibm.com/xml Go Links Norton AntiVirus

Select a country

← developerWorks

alphaWorks emerging technologies

28 September 2003

Contact aW About aW Site map

alphaWorks

Emerging Technologies

- Autonomic Computing
- Collaboration
- Eclipse Technology
- Grid Computing
- Java Technology
- Privacy & Security
- XML
- Web Services
- Wireless

find a technology →

Licensing

Discussion Forums

alphaWorks Newsletters

Newsletters

Subscribe to the aW flash newsletter:

your e-mail

Text HTML

[More newsletters](#)

Free CD, tools, online tutorials & support

Speed-start Web services

NEW! developerWorks

Grid computing zone

Technical resources for developers

alphaWorks > XML

Solution Enabler

[Solution Enabler](#) is a framework for creating and deploying solutions locally or to remote machines with different operating systems. The framework helps to simplify the creation and deployment of software solutions by capturing detailed knowledge of a solution package deployed through a common installer.

NewXML Technologies RSS Feeds **XML**

[Solution Enabler](#) - A framework for creating and deploying solutions locally or to remote machines with different operating systems.

[Web Services for DB2 Cube Views](#) - Web services that allow client applications to perform basic online analytical processing (OLAP) over the Web, using XML and XPath.

[IBM XML Forms Package](#) - A toolkit consisting of software components designed to showcase the possibilities presented by XForms, an emerging W3C standard (W3C Candidate Recommendation).

[SOAP for CICS](#) - A tool that allows integration with SOAP messaging to new and existing CICS Transaction Server applications.

[Struts Action Scripting](#) - A Struts plug-in that allows development of Struts actions using the power and simplicity of any favorite scripting language.

[XincaML](#) - A package including the XincaML language specification and the XincaML Processor, which is a Java™ implementation package of the language parser and constraints checker.

[XML Processing Plus Plus](#) - A typed and stream-based XML processing extension for Java.

[Web Services Outsourcing Manager](#) - A framework that enables dynamic composition of Web service flow based on customer requirements.

[XSLerator](#) - A tool that can generate XSLT scripts from mappings defined using a visual interface.

[More new technologies](#)

Updated XML Technologies

[Emerging Technologies Toolkit](#) - Version 1.1.1 contains a WS-Policy demo, Self-Healing/Optimizing Autonomic Computing demo, Autonomic Computing Toolset, Common Base Event Data Format, Web Services Integration, Web Services Failure Recovery, IBM Grid Toolbox infrastructure along with a Grid Software Manager, WS-Reliable Messaging demo, and a JMX Bridge.

[Web Services Tool Kit for Mobile Devices](#) - Version 2.0.1: Preview implementation of the JSR 172 Version 0.9 specification and support for J2ME, WCE, and SMF environments. C-based Web services has been updated to gSOAP 2.2.3.

[XML Schema Quality Checker](#) - Version 2.2: Fixes from *XML Schema 1.0 Specification Errata* (as of June 1, 2003) implemented; improved detection; use of Xerces-J 2.4; SQC can now run on JRE 1.4.

XML

- All
- Database
- DTDs
- Editors
- Formatters
- Pages
- Programming
- Severs
- Utilities
- XSL

Top XML Downloads

- Emerging Technologies Toolkit
- Xeena
- XML for C++
- XML Viewer
- XML Schema Quality Checker

Development Resources

- developerWorks XML Zone
- Open Source Projects
- XML.org
- The Apache XML Project

Licensable Technologies

- XML Schema Quality Checker
- XML Security Suite
- XML Productivity Kit for Java

Top Online Demos

- Emerging Technologies Toolkit
- WSDL Explorer

alphaWorks Highlights

IBM Adding Eclipse, Grid,

Internet

XML - pretty *HOT* stuff

▶ IBM developerWorks and XML

Product domains

- IBM developer solutions
- DB2
- eServer
- Lotus
- Rational
- Tivoli
- WebSphere
- Toolbox subscription

IBM PartnerWorld

- Benefits of joining
- Technical resources and support
- Products and technologies

Discussion forum

Join the discussion. Ask questions; get advice through an expert-hosted discussion forum. The forums now include several [new functions](#) -- enjoy!

➔ [XML and Java technology](#): Want more on how these two technologies interact? XMLJava technology innovator Brett McLaughlin is here to help.

Columns

XML Watch by Edd Dumbill
New! Edd looks at [Dashboard](#), an open-source application that provides a continuous real-time search of desktop information. (See [previous columns](#).)

XML Matters by David Mertz
Read up on [Text Encoding Initiative \(TEI\)](#), an XML schema devoted to the markup of literary and linguistic texts. (See [previous columns](#).)

Thinking XML by Uche Ogbuji
Uche reviews a new book on [XML Topic Maps](#) and shows you how they can help you get organized. (See [previous columns](#).)

XML for Data by Kevin Williams
Take advantage of the [reusable XML components](#) that Kevin defined in the previous two installments of this column. (See [previous columns](#).)

Working XML by Benoît Marchal
Benoît launches a new project -- an [XML client for e-commerce](#) that can be used by both small and large organizations. (See [previous columns](#).)

Tips

- ➔ New! [Convert from HTML to XML with HTML Tidy](#). (Articles)
- ➔ [When to use local and global declarations](#). (Articles)
- ➔ [Send and receive SOAP messages with JAX-RPC](#). (Articles)
- ➔ [Subscribe to the weekly developerWorks XML tips newsletter](#).
- ➔ [Here's a running list of dW XML tips](#). (Articles)

Most popular links

- ➔ [Tutorial: Understanding DOM](#)
- ➔ [Tutorial: Understanding SAX](#)

Speed-start Web services

News ➔ [More news](#)

- [XML news and analysis from Robin Cover](#) (Oasis)
- [Web services ID management touted](#) (InfoWorld)
- [Tuning your XML for clean Web services](#) (Integration Developer News)
- [The way ahead for Web services](#) (vunet.com)

alphaWorks code

Download and develop with:

- [Web Services for DB2 Cube Views](#)
- [XML for Tables](#)
- [IBM XML Forms Package](#)

From IBM

- Implement [data federation](#) with DB2 Information Integrator.
- [Generate persistent object ID services](#) with WebSphere Studio V5.
- Build metadata bridges for [DB2 UDB Cube Views](#).
- [Machine translation demo](#): Try MT on your favorite Web pages.

Powered by [WebSphere](#)

Done, but with errors on page. Internet

XML - pretty *HOT* stuff

▶ XML.COM is another good location

XML.com: XML From the Inside Out -- XML development, XML resources, XML specifications - Microsoft Internet Explorer provided b

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History

Address <http://www.xml.com/> Go Links Norton AntiVirus

O'REILLY **xml.com**
xml from the inside out

NOW available xmlspy® 2004  www.altova.com

XML.COM WEBSERVICES.XML O'REILLY NETWORK OREILLY.COM

Resources | Buyer's Guide | FAQs | Newsletter | Tech Jobs | Safari Bookshelf search

Get the facts about the most complete all-digital wireless 3G network in the nation.

TOPICS

- Business
- Databases
- Graphics
- Metadata
- Mobile
- Programming
- Schemas
- Style
- Web
- Web Services

XML COLUMN

Practical XQuery
[Ivelin Ivanov on the state of XQuery adoption and future opportunities](#)

XML COLUMN

XML-Deviant
[Kendall Clark reports on the state and trends of XML authoring tools](#)

XML COLUMN

Transforming XML
[Bob DuCharme explains how to use XSLT 2.0's new datatypes system.](#)

Sponsored By:

 www.altova.com

Register Now!

Build with the Best in XML Live Webinar Series

SOFTWARE AG THE XML COMPANY

webservicex.xml.com

- [What is Service-Oriented Architecture?](#) [[webservicex.xml.com](#)]
- [Integrating Services with XSLT](#) [[webservicex.xml.com](#)]
- [A Preview of WS-I Basic Profile 1.1](#) [[webservicex.xml.com](#)]
- [Using XPath with SOAP](#) [[webservicex.xml.com](#)]

Marking Up Bureaucracy
by Paul Ford
Needing to cope with its enormous needs for document and data exchange, the United States is looking more and more to XML. Paul Ford explains what happens when Washington meets markup.

ISO to Require Royalties?
by Kendall Grant Clark
The ISO, a worldwide standards body, is proposing to charge fees for commercial usage in software of their standardized country, language and currency codes. This would have a wide-ranging negative effect on the infrastructure of the web and related standards. Kendall Grant Clark explains the situation and argues against the ISO's proposal.

Web Disservices: Microsoft's Misstep

Done Internet

▶ XML.COM is another good location

XML.com: XML From the Inside Out -- XML development, XML resources, XML specifications - Microsoft Internet Explorer provided b

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Mail

Address <http://www.xml.com/> Go Links Norton AntiVirus

ESSENTIALS

- [Annotated XML](#)
- [What is XML?](#)
- [What is XSLT?](#)
- [What is XSL-FO?](#)
- [What is XLink?](#)
- [What is XML Schema?](#)
- [What is XQuery?](#)
- [What is RDF?](#)
- [What is RSS?](#)
- [What are Topic Maps?](#)
- [What are Web Services?](#)
- [What are XForms?](#)
- [XSLT Recipe of the Day](#)

[Manage Your Account](#)
[Forgot Your Password?](#)

FIND

- [Search](#)
- [Article Archive](#)
- [FAQs](#)

Add the latest news from XML.com to your web site
[Click here!](#)

COLUMNS

- [XML-Deviant](#)
- [Style Matters](#)
- [XML Q&A](#)
- [Transforming XML](#)
- [Perl and XML](#)
- [Rich Salz](#)

by Mark Pilgrim
In this month's Dive Into XML column, Mark Pilgrim takes a look at Microsoft's new Microsoft.com web service, suggesting that it might be improved by becoming more like the Web itself.

An Introduction to STAX
by Elliott Rusty Harold
STAX, the Streaming API for XML, is a new API for pull-parsing of XML, developed under the Java Community Process as JSR 173. O'Reilly author Elliott Rusty Harold gives an introduction to this API, which combines the efficiency of SAX with the ease of use of tree-based APIs.

Language Instincts
by Jon Udell
There'll be no master plan to the Semantic Web, says Jon Udell, just a lot of talking, listening and imitating.

An XQuery Update
by Per Bothner
A report on the changes made to the W3C's XML Query Language in the recent August 2003 XQuery drafts.

The State of the Python-XML Art, 2003
by Uche Ogbuji
In this month's Python and XML column Uche Ogbuji updates his report on the state of the Python-XML art, adding 24 new projects.

Ten Favorite XForms Engines
by Micah Dubinko
The author of O'Reilly's "XForms Essentials" describes ten software packages that implement the W3C's XForms specification, seen as the XML-friendly successor to HTML forms.

Writing Your Own Functions in XSLT 2.0
by Bob DuCharme
In this month's Transforming XML column Bob DuCharme explains how to write arbitrary XSLT functions in XSLT 2.0.

Web Services News

- [Actional, AmberPoint, and Empirix Announce New Web Services Management Tools \[WebServices.org\]](#)
- [Financial Services Technology Consortium Issues Results of Web Services Proof of Concept \[WebServices.org\]](#)
- [Layer 7 Technologies Introduce Security And Policy Coordination Solution For Web Services \[WebServices.org\]](#)
- [OASIS Forms Technical Committee to Formulate Web Services Implementation Guidelines \[WebServices.org\]](#)
- [MS Mobiles Release Free Web Services Client for Microsoft Smartphone 2003 \[WebServices.org\]](#)
- [HASH\(0x854fe0c\) \[WebServices.org\]](#)
- [Web Services Security Company Forum Systems On Track With Funding \[WebServices.org\]](#)
- [IBM Release Updated Version of WebSphere SDK for Web Services \[WebServices.org\]](#)

Events

- [Business Integration & Web Services Conference Series \[Nov. 4, 2003\]](#)

XML News

Palm OS Version 4.1
Only 4" Tall
Less Than 1" Thick

XML - pretty *HOT* stuff

- ▶ **This site has excellent starting tutorials!**

W3Schools Online Web Tutorials - Microsoft Internet Explorer provided by AT&T WorldNet Service

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Mail

Address <http://www.w3schools.com/> Go Links Norton AntiVirus

W3 Schools

The best things in life are free

HTML Tutorials

- Learn HTML
- Learn XHTML
- Learn CSS

XML Tutorials

- Learn XML
- Learn XSL
- Learn DTD
- Learn DOM
- Learn WAP
- Learn XSD
- Learn XPath
- Learn XForms
- Learn SOAP
- Learn WSDL
- Learn XMLEspy

Browser Scripting

- Learn JavaScript
- Learn DHTML
- Learn VBScript
- Learn WMLScript

Server Scripting

- Learn SQL
- Learn ASP
- Learn ADO
- Learn PHP

.NET (dotnet)

- .NET Microsoft
- .NET ASP
- .NET Mobile

Multimedia

- Learn Media
- Learn SMIL
- Learn Flash

Web Building

W3Schools - Full Web Building Tutorials - All Free

At W3Schools you will find all the Web-building tutorials you need, from basic HTML and XHTML to advanced XML, XSL, Multimedia and WAP.

W3Schools - The Largest Web Developers Site On The Net

Full Web Building References

Our references cover all Web-building technologies, including W3C standards like HTML, XHTML, CSS, XML and other technologies like ASP and ADO plus much more.

Try-It-Yourself On-Line Examples

At W3schools you will find thousands of cut-and-paste examples. Our on-line HTML editor will let you edit the examples and experiment with the code on-line.

Quizzes - Test Your Skills

With our quizzes you can test your skills and prepare yourself for professional work.

About W3Schools

REFERENCES

- [HTML 4.01](#)
- [XHTML 1.0](#)
- [CSS 2.0](#)
- [XSLT 1.0](#)
- [WML 1.1](#)
- [ASCII Reference](#)
- [Entity Reference](#)
- [HTML Colors](#)
- [Color Names](#)
- [Color Values](#)
- [WAP Pages](#)

QUIZZES

- [HTML Quiz](#)
- [XHTML Quiz](#)
- [CSS Quiz](#)
- [XML Quiz](#)
- [JavaScript Quiz](#)
- [ASP Quiz](#)
- [SQL Quiz](#)

EXAMPLES

- [HTML Examples](#)
- [CSS Examples](#)
- [XML Examples](#)
- [DOM Examples](#)
- [WAP Examples](#)
- [JavaScript Examples](#)
- [DHTML Examples](#)
- [VBScript Examples](#)

Done Internet

XML - pretty *HOT* stuff

▶ Try these good starting tutorials!

HOME

- XSL Tutorial**
- XSL HOME
- XSL Introduction
- XSL Languages
- XSL Browsers
- XSL Transform
- XSL template
- XSL value-of
- XSL for-each
- XSL sort
- XSL if
- XSL choose
- XSL apply-templates
- XSL on the Client
- XSL on the Server

References

- XSLT Elements
- XSLT Functions

XSL Tutorial

Previous Next

XSL Tutorial

In our XSL tutorial you will learn what XSL is. You will also learn how to use XSL to transform XML documents into other formats, like HTML.

[Start learning XSL!](#)

XSL References

At W3Schools you will find complete XSL references about both XSL elements and their attributes.

[XSL References](#)

Table of Contents

- [XSL Introduction](#)
An introduction to XSL - The style sheet language of XML. What XSL is and what it can do.
- [XSL Languages](#)
Defines the sub-languages of XSL: XSLT, XPath and XSL Formatting Objects.
- [XSL Browsers](#)
Overview of the XSL browser support.
- [XSL Transformation](#)
How XSL can be used to transform XML documents into XHTML documents.
- [XSL <xsl:template>](#)
The <xsl:template> element contains rules to apply when a specified node is matched.
- [XSL <xsl:include>](#)

ECOMMERCE COMPONENTS

[ASP Ecommerce & Shopping Cart](#)

Stylesheet Editor

Powerful Stylesheet Editor
Syntax Help, Easy-to-use, Free D/L.
[www.xmlspy.com](#)

Stylesheet samples, tests

Free XSL tutorial.
XML - XSL to PDF
Free trial. Cross platform, standard
[www.renderx.com](#)

XML XSL Tool

Visually Generate XML from Diagrams
Try VisualScript Free!
[www.smartdraw.com](#)

Done Internet

- There is so much going on with the XML world you just have to keep checking out the www.w3.org site to see what the latest is

- But.....you should have at least a working starting knowledge and be able to look at XML documents and understand them as well as DTD, SCHEMA, etc....etc....
 - *XML is really just taking off*

 - ***SO KEEP YOUR SEAT BELTS FASTENED!***