Styling XML

Hans C. Arents

senior IT market analyst

I.T. Works

"Guiding the IT Professional"

Innovation Center, Technologiepark 3, B-9052 Gent (Belgium), Tel: +32 (0)9 241 56 21 - Fax: +32 (0)9 241 56 56

E-mail: hca@itworks.be - Site: http://www.itworks.be/ - Home: http://www.arents.be/

Styling XML

- Part 1: Transforming XML 13u30 – 16u00

  coffee break

- Part 2: **Displaying XML** 16u30 – 17u30

  questions & answers

Displaying XML

Hans C. Arents

senior IT market analyst

I.T. Works

"Guiding the IT Professional"

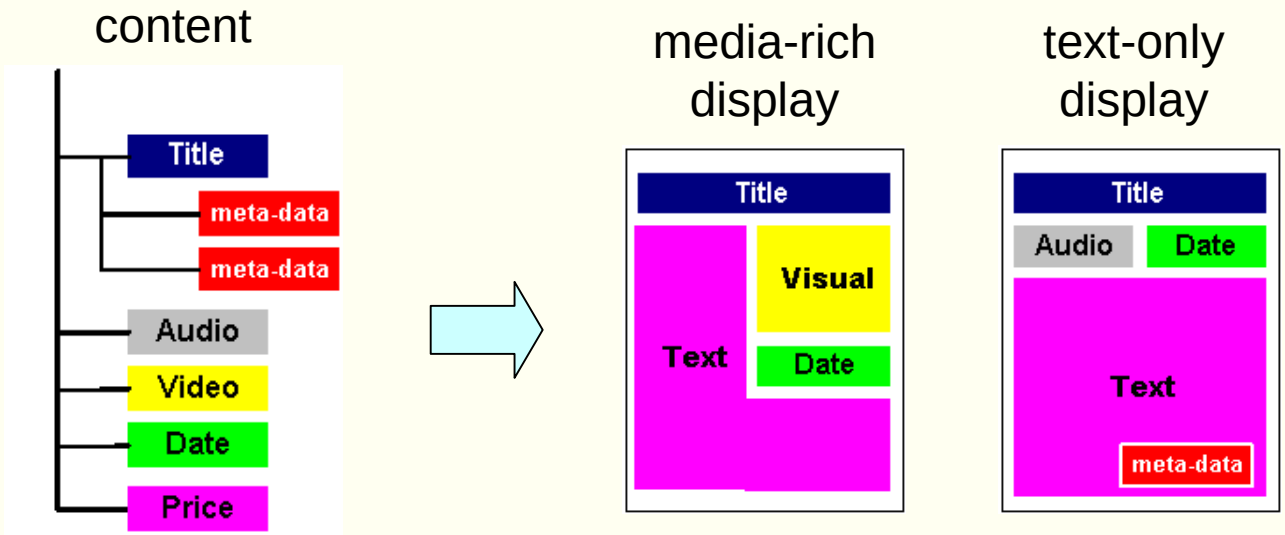
Innovation Center, Technologiepark 3, B-9052 Gent (Belgium), Tel: +32 (0)9 241 56 21 - Fax: +32 (0)9 241 56 56

E-mail: hca@itworks.be - Site: http://www.itworks.be/ - Home: http://www.arents.be/

Displaying XML

- What are the challenges?
- How to display XML in a Web browser
  - CSS vs. XSL stylesheets
  - when to use CSS / not to use XSL
- Transforming XML to HTML
  - using XSL client-side
  - using XSL server-side
    - ASP pages
    - XSL ISAPI
- Tools overview
  - XML viewers
  - XSLT processors
  - XSL editors

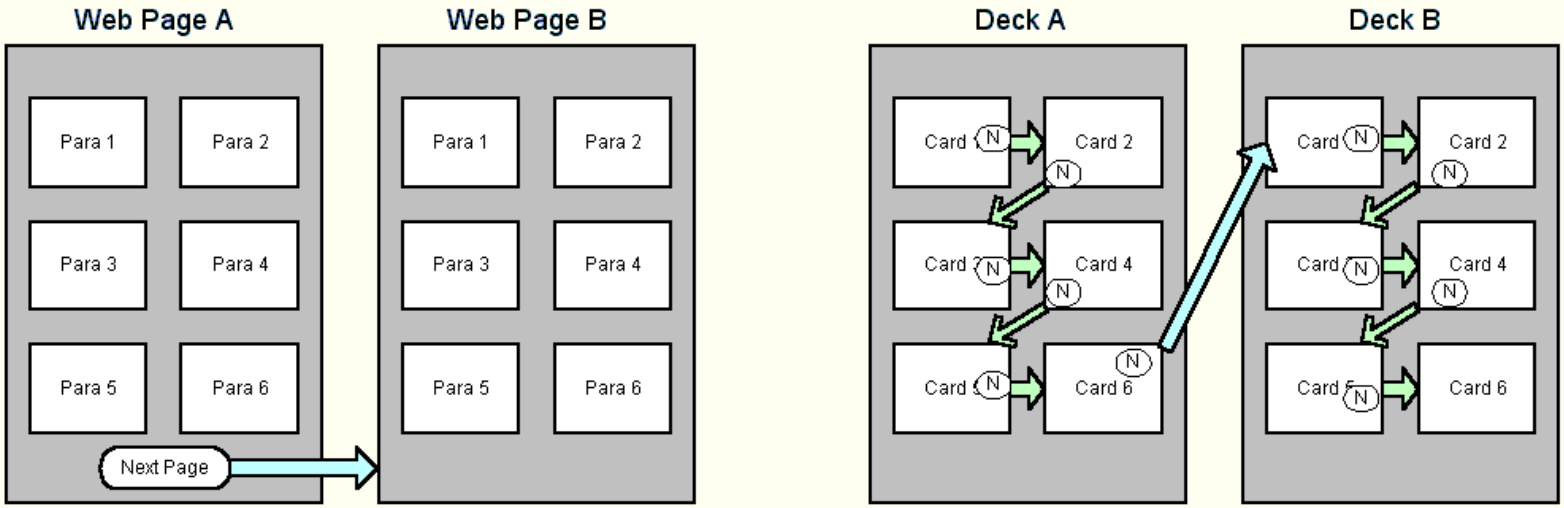
Separating content and presentation



- Semantically rich content using XML
  - knowledge about the meaning of the data, datatypes, metadata

  presented in different ways ("look") using XSL
  - restyling the presentation
  - re-using the content
- Goal: *write once and display anywhere*

Separating content and navigation



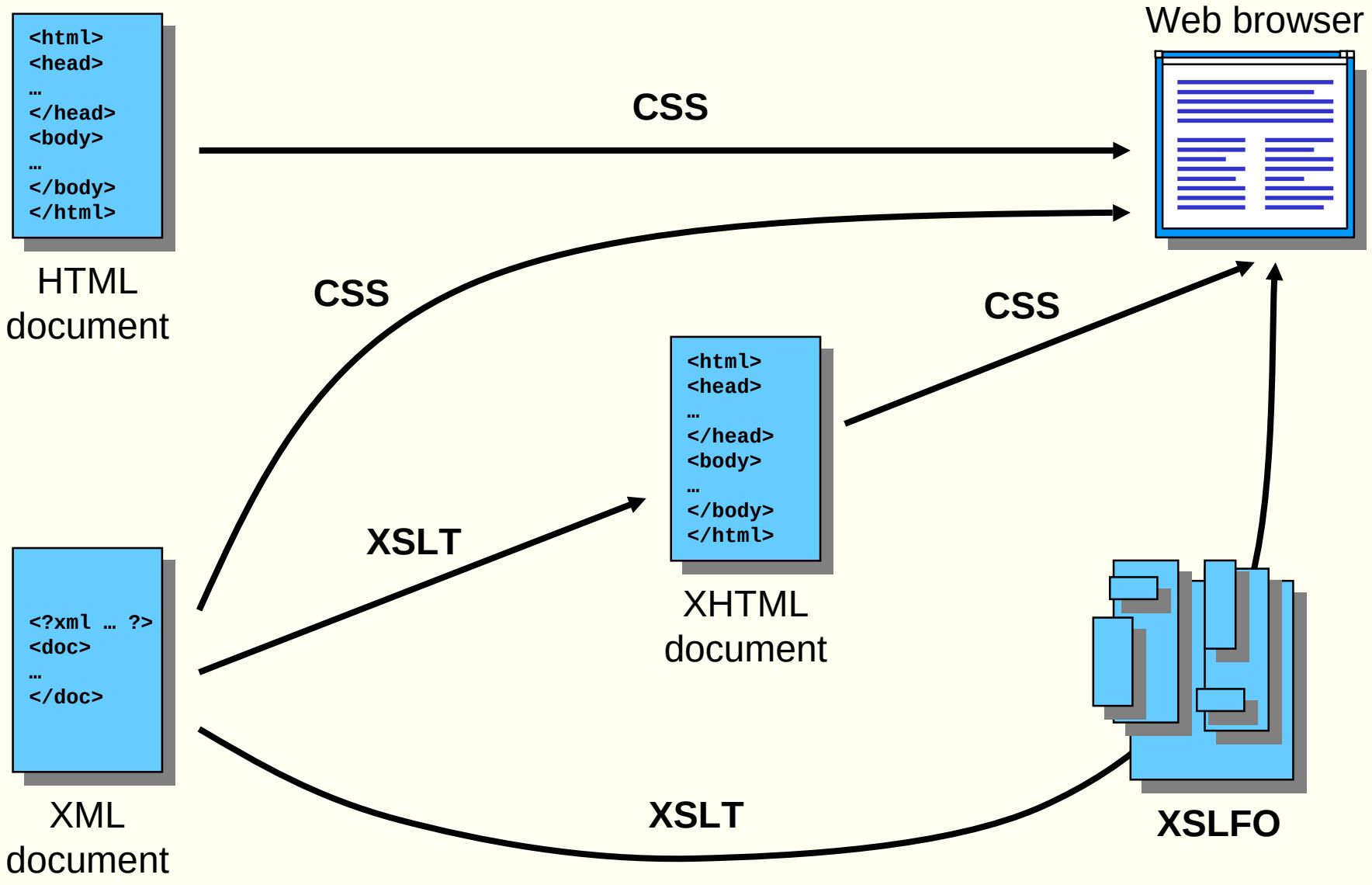
- Same rich document content using XML adapted to different form factors ("feel") using XSL
  - different content chunking
    - e.g. page vs. paragraph granularity
  - different navigation behaviour
    - e.g. random vs. conversational access
- Goal: *write once and navigate anywhere*

Targeting different access devices

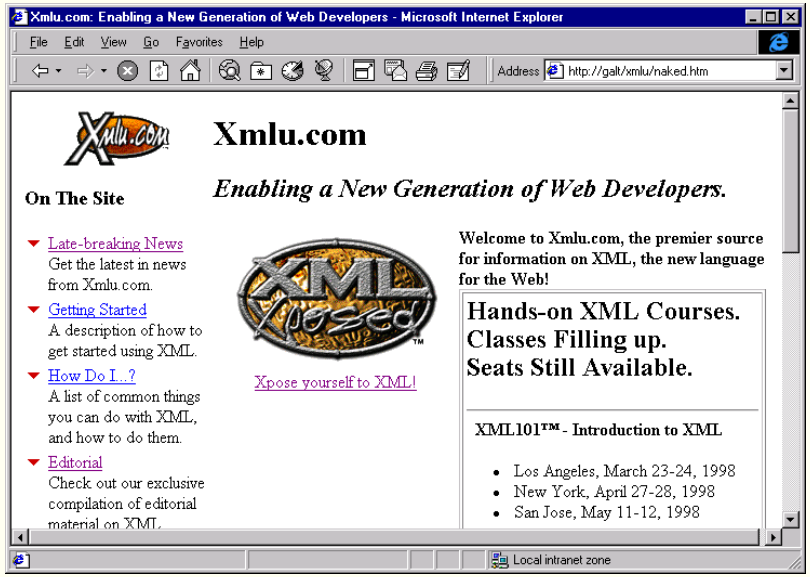
fixed PC | mobile PC | handheld PC | mobile phone | voice | WebTV | WebPad

- by 2002: 75% of all Internet access using non-PC devices
- never-ending need to adapt and optimize presentation
  - different display markup languages:
    - fixed and mobile PC: HTML, XHTML / handheld PC: HDML
    - mobile phone: WML (WAP), cHTML (i-mode) / voice: VoxML
  - different device characteristics:
    - different sizes of screens / different screen resolutions
    - new ways of interacting (e.g. "callto" hyperlink, talkback button)

How to display XML in a Web browser



CSS (Cascading Style Sheets)

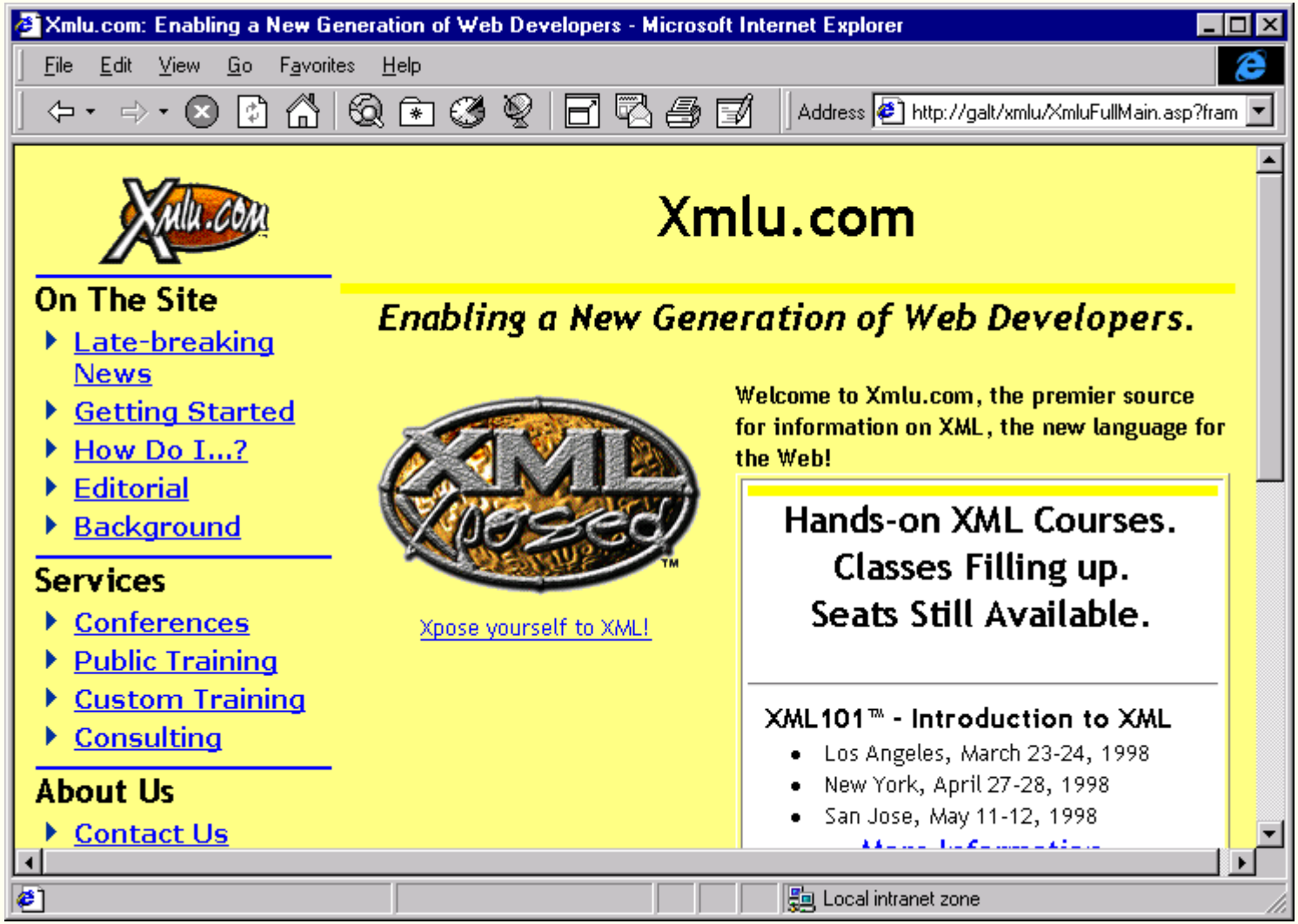


+

```
<style type='text/css'>
  BODY       {font-family:Trebuchet MS,Verdana,Arial;
              color:black;
              background-color:#FFFF80}
  A:link     {color: blue;}
  A:visited  {color: blue;}
  A:hover    {color: red;}
  center     {text-align:center;}
  TD         {font-size:10pt;bottom-margin:0;
              top-margin:0;}
  LI         {margin-left:0}
  H1         {text-align:center;
              font-size:24pt;
              font-weight:bold;}
  H2         {text-align:center;
              font-size:16pt;
              font-weight:bold;
              border-top:thick yellow solid;}
  H3         {font-size:14pt;
              font-weight:bold;
              border-top:thin blue solid;
              margin-bottom:0;
              margin-top:5}
</style>
```

=

CSS (Cascading Style Sheets)



CSS vs. XSL

- CSS (**C**ascading **S**tyle **S**heets)
  - one-to-one mapping between an XML element and a built-in browser formatting object
    "sprinkling presentation hints on top of XML"
  - structure of output formatted document closely follows structure of input XML document
  - client-side only styling
- XSL (**E**xtensible **S**tylesheet **L**anguage)
  - XML element mapped to a formatting object consisting of multiple built-in formatting objects
    "defining detailed presentation using XML"
  - structure of output formatted document is not constrained by structure of input XML document
  - client-side and server-side styling

CSS vs. XSL

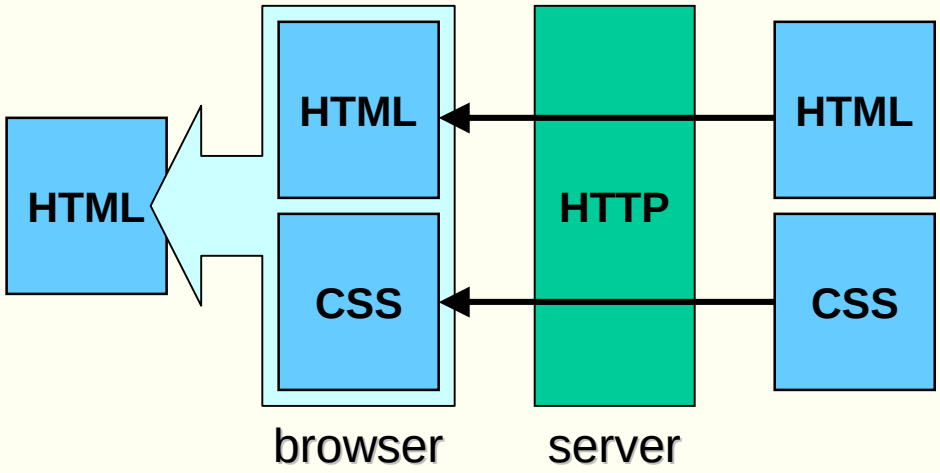
	CSS	XSL
Can be used with HTML?	Yes	No
Can be used with XML?	Yes	Yes
Is a transformation language?	No	Yes
Uses which syntax?	Own syntax	XML syntax

When to use CSS / not to use XSL

- When to use CSS
  - for transformations aimed at rendering
  - when the structure of the source document is very close to the structure of the target display format
  - if you don't need to reorder and restructure the information or calculate/generate new information
- When not to use XSL
  - for up-translations
    - going from a less structured to a more structured format
  - for transformations based on content
    - using the content of elements/attributes rather than the tree structure
  - for transformations of directed graphs encoded using XML
    - bad for manipulating graph-structured data, good for tree-structured data
- Advice: for XML **à** HTML, use XSL first, followed by CSS

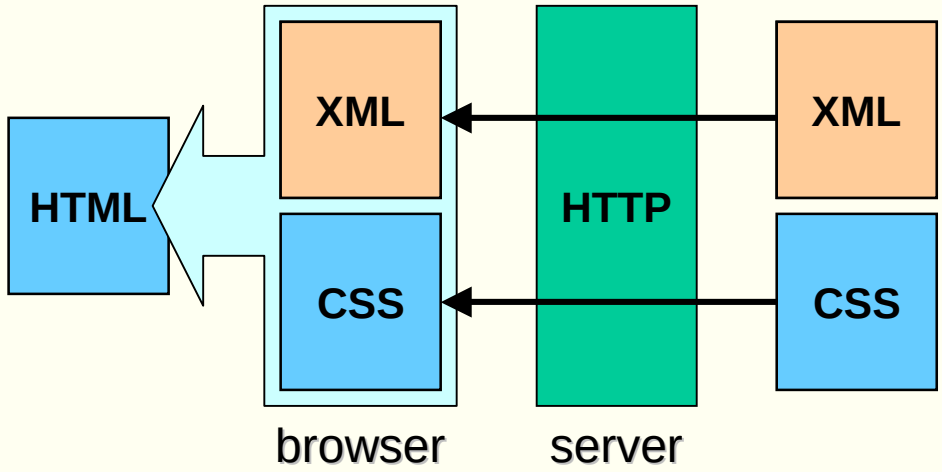
Transforming XML to HTML: client-side

- CSS: client-side processing only

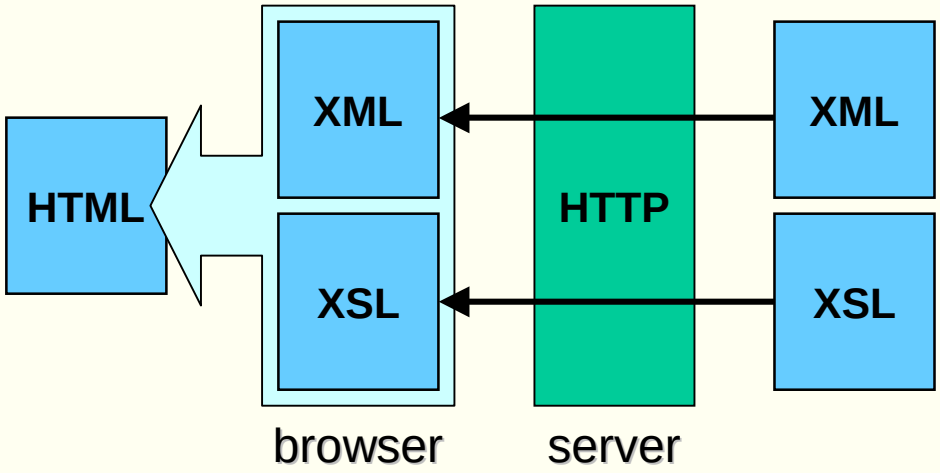


Transforming XML to HTML: client-side

- CSS: client-side processing only



- XSL: client-side processing



Transforming XML to HTML: client-side

XML input file

```
<?xml version="1.0"?>
<?xml-stylesheet href="books2html.xsl" type="text/xsl"?>
<booklist>
  <book>
    <title>ADO - The Savior of Data Access</title>
    <isbn>123-009923-13</isbn>
    <publisher>Makes No Sense Press</publisher>
    <author>Kenny Loggins</author>
    <author>Eric Cartman</author>
  </book>
  <book>
    <title>XML for Slightly Dumb People</title>
    <isbn>123-009923-12</isbn>
    <publisher>Really Simple Books</publisher>
    <author>Bobby Schmidt</author>
    <author>Malcolm XML</author>
  </book>
  <book>
    <title>So You Want To Be Broke investing in IT?</title>
    <isbn>123-009923-14</isbn>
    <publisher>Financial Wizards</publisher>
    <author>Timmy Two Time</author>
    <author>John Deadmeat</author>
  </book>
</booklist>
```

`xml-stylesheet` processing instruction

- Purpose: load the desired stylesheet and perform the transformation on the XML source tree
  - argument `href` to specify the URI of the stylesheet
    - XSL stylesheets typically have the extension `.xsl`
  - argument `type` to specify the type of stylesheet to use
- Syntax:

```
<?xml-stylesheet href="file-name" type="text/xsl"?>
```

  - an XSL stylesheet with name `file-name` is to be executed
- Similar syntax for CSS stylesheets:

```
<?xml-stylesheet href="file-name" type="text/css"?>
```

Transforming XML to HTML: client-side

XSL input file (MS-specific syntax!)

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <HTML>
      <TABLE WIDTH="100%" BORDER="1">
        <TR>
          <TH>Title</TH><TH>Authors</TH><TH>Publisher</TH><TH>ISBN</TH>
        </TR>
        <xsl:apply-templates select="//book" order-by="-title"/>
      </TABLE>
    </HTML>
  </xsl:template>
  <xsl:template match="book">
    <TR>
      <TD><xsl:value-of select="title"/></TD>
      <TD>
        <xsl:for-each select="author">
          <xsl:value-of select="."/><br/>
        </xsl:for-each>
      </TD>
      <TD><xsl:value-of select="publisher"/></TD>
      <TD><xsl:value-of select="isbn"/></TD>
    </TR>
  </xsl:template>
</xsl:stylesheet>
```

Note the non-standard MS-specific XSLT syntax!

Why this MS-specific XSLT syntax?

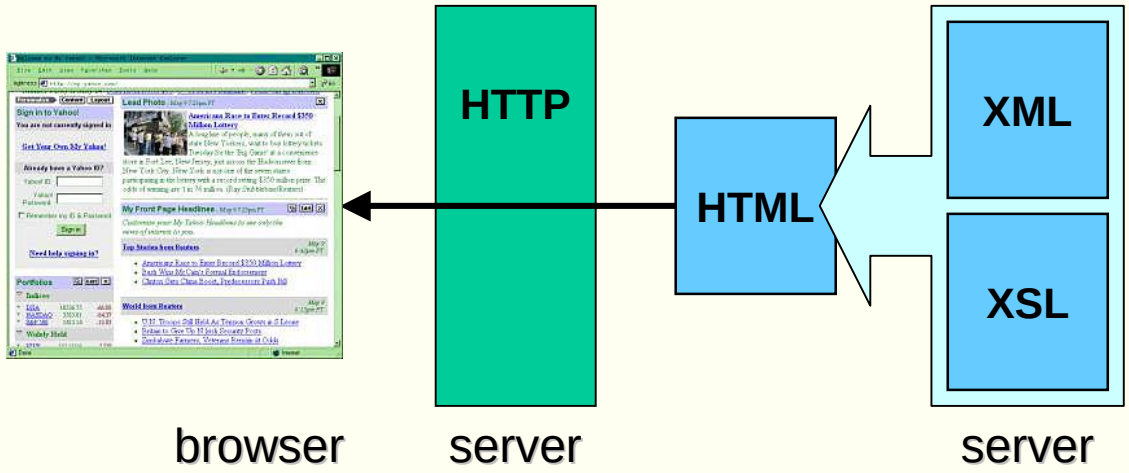
- XSL functionality of IE provided by the MS XML parser
  - MSXML DLL = XML parser + XSL processor
- Different versions of the MSXML DLL:
  - Microsoft *MSXML* (IE 5.01 version) to be used in ASP, VB, …
    - warning: supports pre-standard version of XSL (“MS-XSL”)
  - Microsoft *MSXML Version 3.0* (October 2000)
    - completely supports standard XSLT and XPath
    - optimizations for improved document throughput (2/3x faster)
      - e.g. server-side XSL stylesheet caching
    - can be installed alongside or as a replacement of the original *MSXML*
  - Microsoft *MSXML Version 4.0* (somewhere in 2001) for use in .NET
    - support for XML Schemas, XML query
- **http://www.netcrucible.com/xslt/msxml-faq.htm**

Transforming XML to HTML: client-side

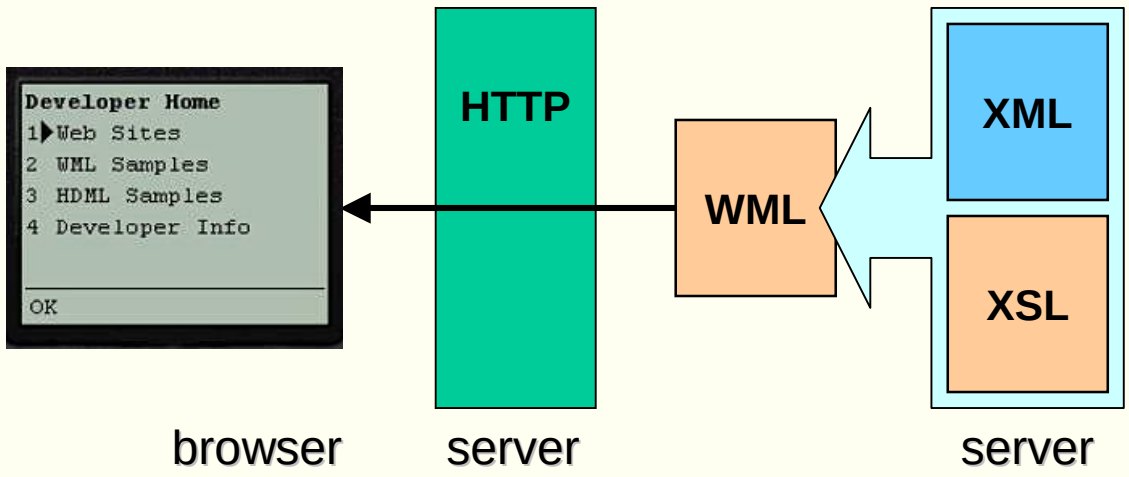
- Microsoft *Internet Explorer 5.5*
  - partial support for CSS 1.0, no support for CSS 2.0
  - built-in MSXML parser: support for "MS XSL" (XSLT working draft)
  - MSXML 3.0 parser: complete support for XSLT standard
  - proprietary extensions:
    - proprietary support for XML Data Islands in HTML pages accessible through DSO (Data Source Objects) or the DOM
- Netscape *Navigator 6.0* / Mozilla 0.8
  - full support for CSS 1.0, partial support for CSS 2.0
  - no XSL support, but XSL processor can be plugged in (Mozilla 0.9?)
    - based on open source XSL processor *TransforMiiX*
- Opera Software *Opera 5.02*
  - full support for CSS 1.0, good support for CSS 2.0
  - no XSL support, and no plans for the future

Transforming XML to HTML: server-side

- XSL: server-side processing

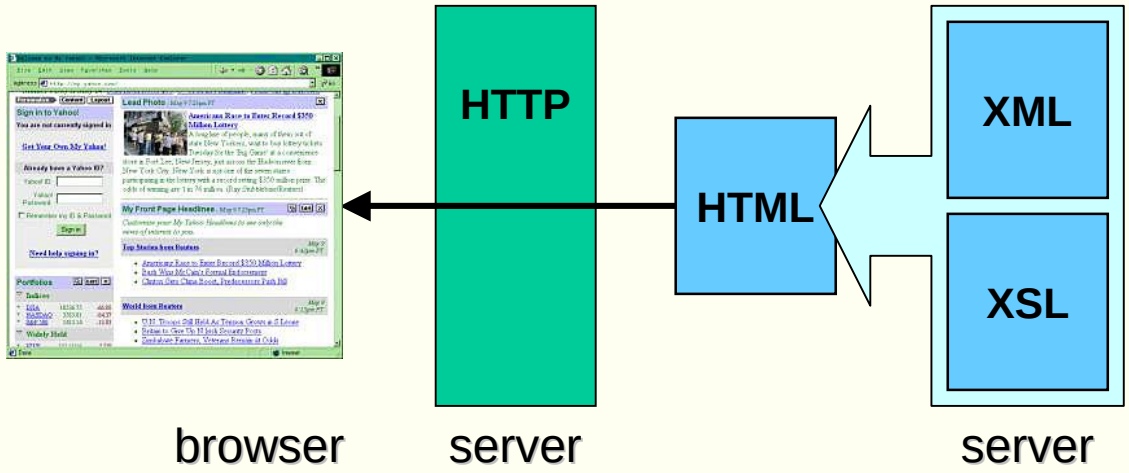


- XSL: server-side processing

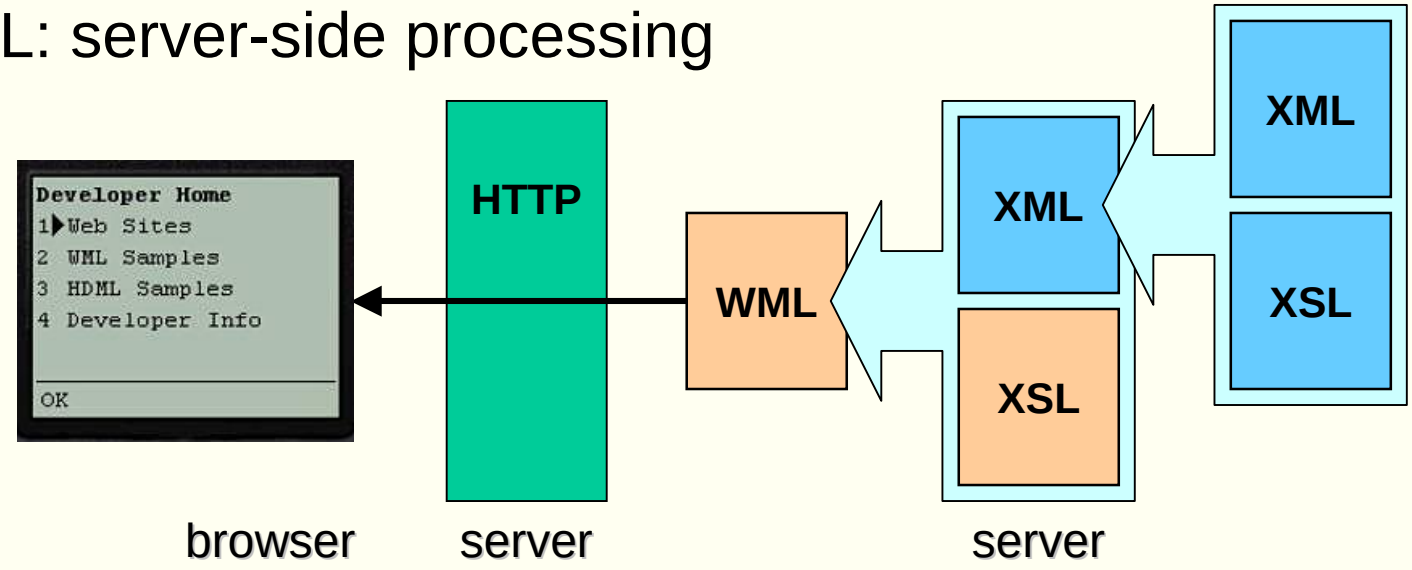


Transforming XML to HTML: server-side

- XSL: server-side processing



- XSL: server-side processing



Transforming XML to HTML: server-side

ASP page

```
<%@ Language=VBScript %>
<%
  Dim oDoc
  Set oDoc = Server.CreateObject("Msxml2.DOMDocument")
  oDoc.load(Server.MapPath("books.xml"))

  Dim oXsl
  Set oXsl = Server.CreateObject("Msxml2.DOMDocument")
  oXsl.load(Server.MapPath("books2html.xsl"))

  Dim oHtml
  oDoc.transformNodeToObject oXsl, Response
%>
```

- Create a DOM tree object and load the XML document into it
- Create a DOM tree object and load the XSL stylesheet into it
- Perform the XSL transformation on the XML source DOM tree
- Send the result back as an HTTP response object

Transforming XML to HTML: client/server-side

ASP page with browser sniffing

```
<%@ Language=VBScript %>
<%
  Dim IsIE
  IsIE = InStr(1,Request.ServerVariables("HTTP_USER_AGENT"),"MSIE") > 0
  If IsIE Then
    Dim fso, file
    Set fso = Server.CreateObject("Scripting.FileSystemObject")
    Set file = fso.OpenTextFile(Server.MapPath("books.xml"))
    Response.ContentType = "text/xml"
    Response.Write file.ReadAll
    file.Close
  Else
    Dim oDoc
    Set oDoc = Server.CreateObject("Msxml2.DOMDocument")
    oDoc.load(Server.MapPath("Books.Xml"))

    Dim oXsl
    Set oXsl = Server.CreateObject("Msxml2.DOMDocument")
    oXsl.load(Server.MapPath("QueryResult.xsl"))

    Dim oHtml
    oDoc.transformNodeToObject oXsl, Response

  End If
%>
```

Transforming XML to HTML: client/server-side

Œ Detect whether the browser requesting the ASP page is a Microsoft Internet Explorer or not

- If it is IE, then the stylesheet can be executed client-side, so …

Ž create a file system object and read the XML document into it

- set the content type of the HTTP response object to XML
- write the XML document into the HTTP response object
  - note: this XML document has to contain an `xml-stylesheet` directive!

```
<?xml-stylesheet href="file-name" type="text/xsl"?>
```

‘ send the HTTP response object back to the browser

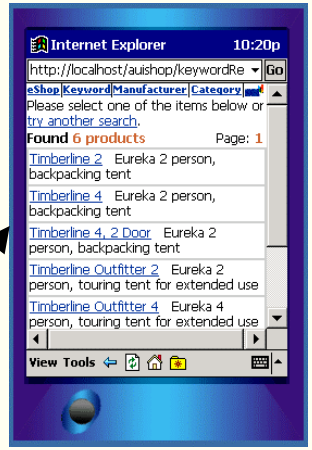
’ If it’s not IE, then the stylesheet has to be executed server-side

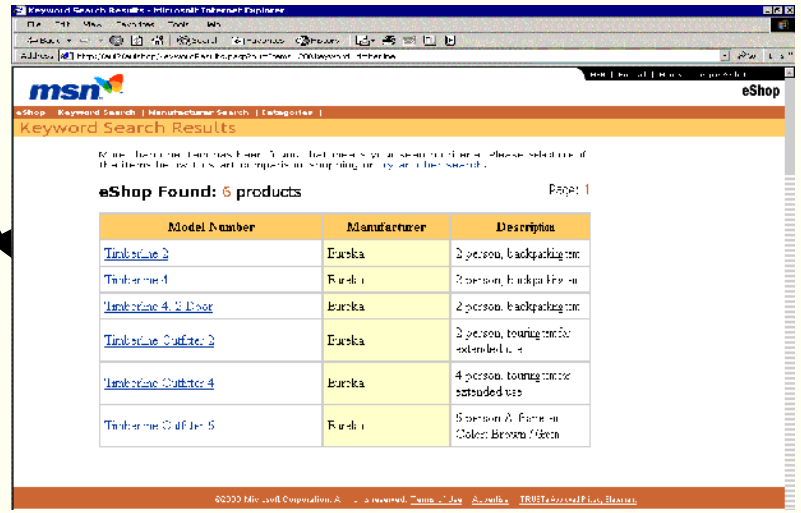
Transforming XML to HTML: server-side

- XSL ISAPI 2.1 (XSL Internet Server Application Programming Interface)
  http://msdn.microsoft.com/xml/general/xslisapifilter.asp
  - filter for Internet Information Server (technology preview)
  - why was this developed?
    - not all client browsers support XSL (e.g. wireless devices)
    - choosing stylesheets based on browser device characteristics, and then executing those stylesheets on the server is difficult
    - dynamically generating XML by using ASP and having the result transformed by XSL on the server is a difficult and heavy process
  - what features does it offer?
    - automatic execution of XSL stylesheets on the server, choosing alternate stylesheets based on browser device characteristics or XML document type
    - configurable "pass-through" for subsequent client-side XSL processing
    - pipelining of ASP-generated XML into XSL stylesheets
    - stylesheet caching for improved server performance
    - chaining execution of multiple stylesheets

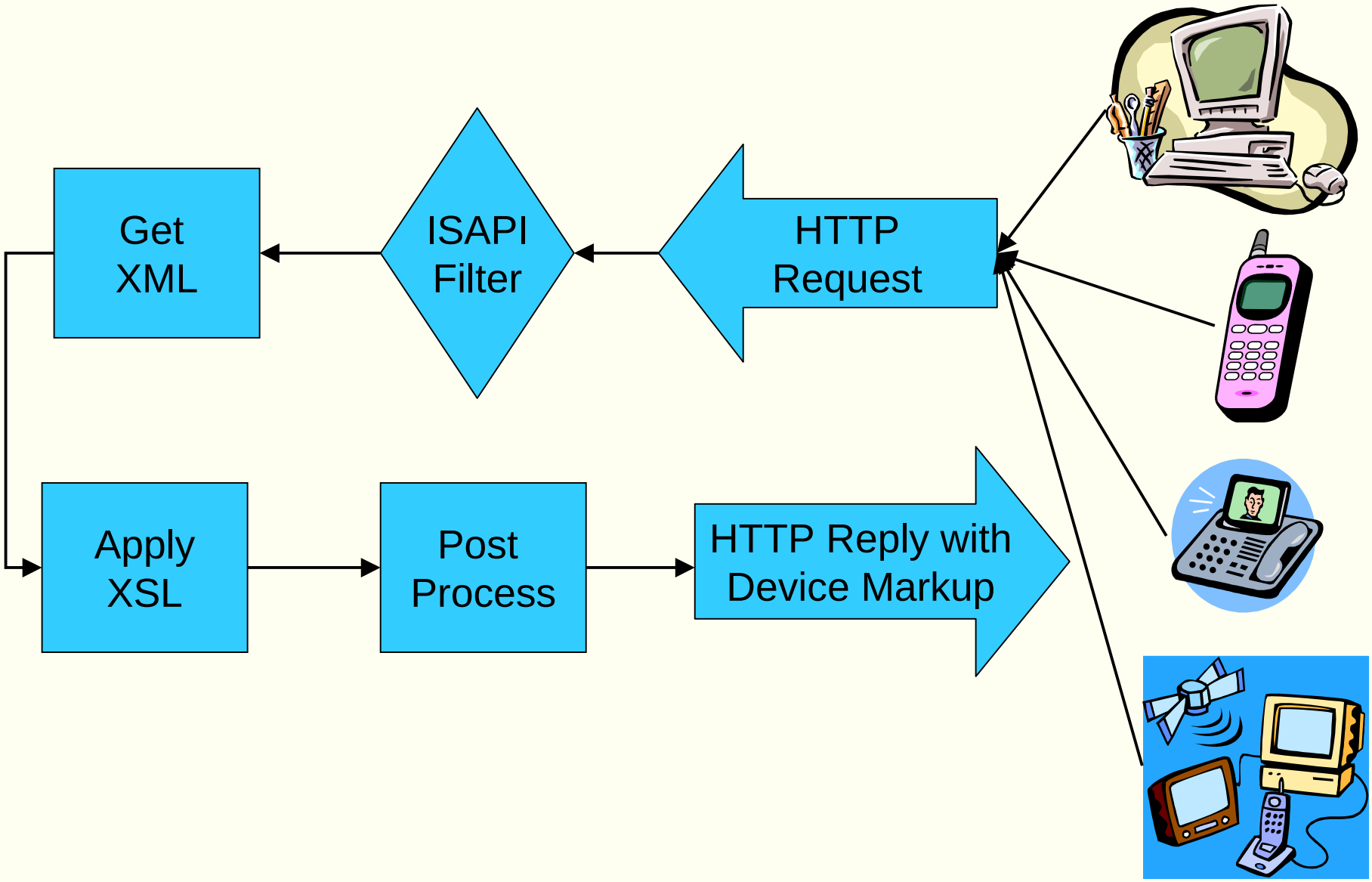
XSL ISAPI 2.1: what does it do?

XSL
ISAPI





XSL ISAPI 2.1: how does it work?



XSL ISAPI 2.1: how does it work?

In the XML page:

```
<?xml-stylesheet type="text/xsl"
                 server-config="myConfig.xml"
                 href="clientStyle.xsl"?>
```

- once the XSL ISAPI filter is installed, the filter will examine all requests for a URL with an **`.xml`** extension
- XML files that have a processing instruction of the above form are selected for processing by XSL ISAPI

XSL ISAPI 2.1: how does it work?

myConfig.xml (with browser-based selection)

```
<?xml version="1.0"?>
<server-styles-config>
  <device browser="IE" version="5.0">
    <stylesheet href="ie50Style.xsl"/>
  </device>
  <device browser="IE" version="3.0">
    <stylesheet href="ie30Style.xsl"/>
  </device>
  <device>
    <stylesheet href="anyStyle.xsl"/>
  </device>
</server-styles-config>
```

- the **<device>** elements are evaluated from top to bottom, with the first match winning, and its stylesheet child elements are selected
- a **<device>** element without any attributes is an automatic match
- if no **<device>** elements match, the XML is sent untouched to the browser, with a content-type of **text/xml è** the **clientStyle.xsl** stylesheet will be retrieved and applied in the client browser

XSL ISAPI 2.1: how does it work?

Doctype-based selection:

```
<device browser="IE" version="5.0">
  <doctype name="invoice">
    <stylesheet href="ie5Invoice.xsl"/>
  </doctype>
  <doctype name="productDescription, description">
    <stylesheet href="ie5Description.xsl"/>
  </doctype>
</device>
```

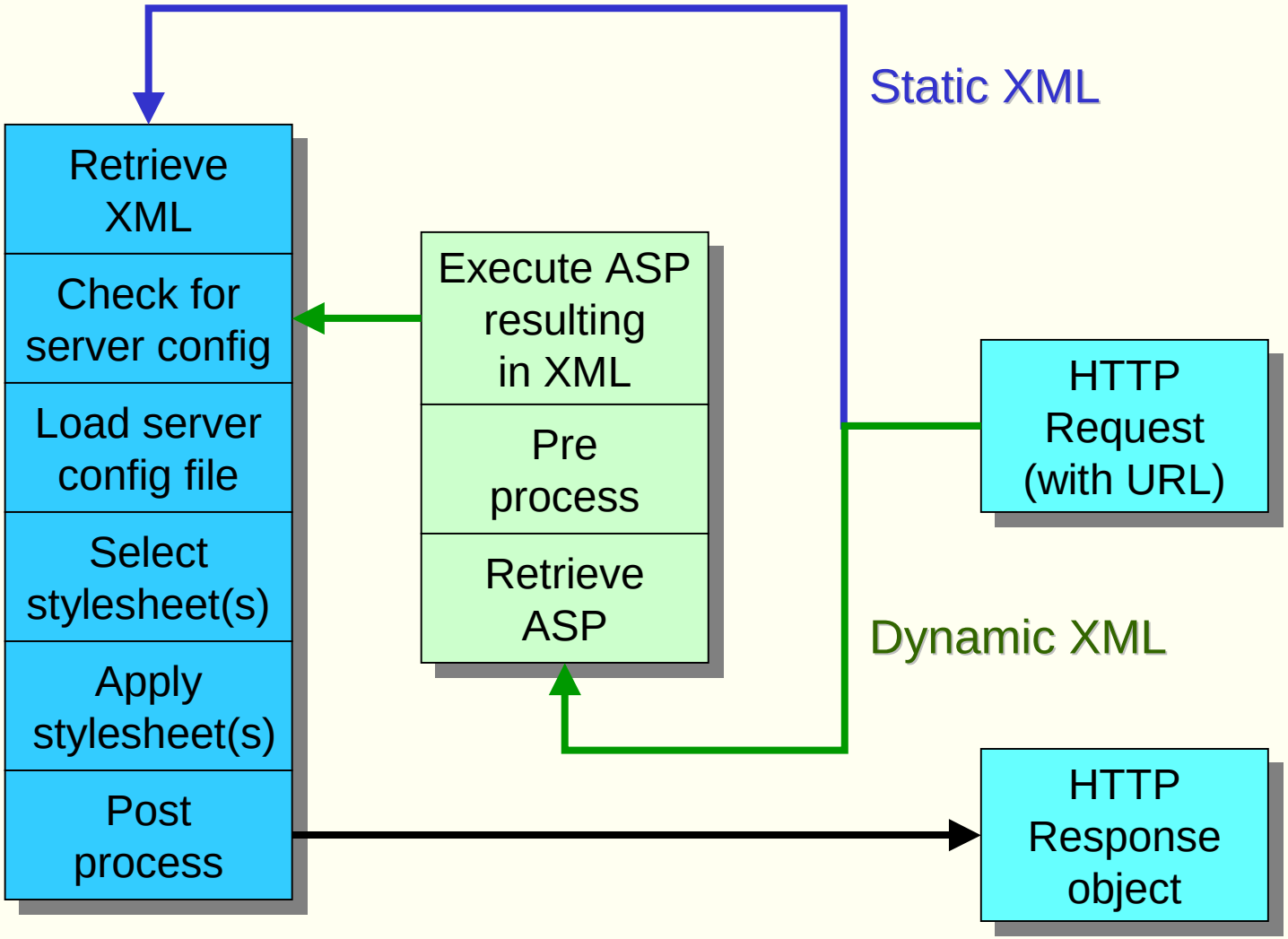
- selects the stylesheet based on the doctype of the XML document

Stylesheet chaining:

```
<device browser="IE" version="5.0">
  <stylesheet href="removeApples.xsl"/>
  <stylesheet href="sortByPrice.xsl"/>
  <stylesheet href="ie50Style.xsl"/>
</device>
```

- allows multiple XSL stylesheets to be chained together
  - all but the last stylesheet must be an XML to XML transformation
  - note: maximum limit of 64 chained stylesheets

XSL ISAPI 2.1: how does it work?



XSL ISAPI 2.1: how does it work?

PASP page:

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" server-config="myConfig.xml"?>
<currentTime>
<%= Now %>
</currentTime>
```

which is converted to:

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" server-config="myConfig.xml"?>
<currentTime>
4/28/2000 2:49:59 PM
</currentTime>
```

- build a content pipeline, not by using static XML
  but by using dynamic XML generated by an ASP page
- ASP script needs to be stored in a file with a **`.pasp`** extension,
  and it must result in a well-formed XML document

XSL ISAPI 2.1

- What about performance?
  - faster than functionally equivalent ASP pages
  - non-XML (e.g. HTML) or ASP files are unaffected
  - filter slows performance of XML files that are not transformed
  - can be considerably faster (up to 50%) with MSXML 3.0 installed
  - recommend scenario: run some performance tests before deploying
- What do I need to use it?
  - IIS (Internet Information Server) 5.0
  - Windows 2000 Server (or Advanced Server)
  - MSXML 3.0 (msxml3.dll) or MSXML 2.5 (msxml.dll)
  - NT 4.0 + IIS 4.0 does work, but with some restrictions and performance degradation

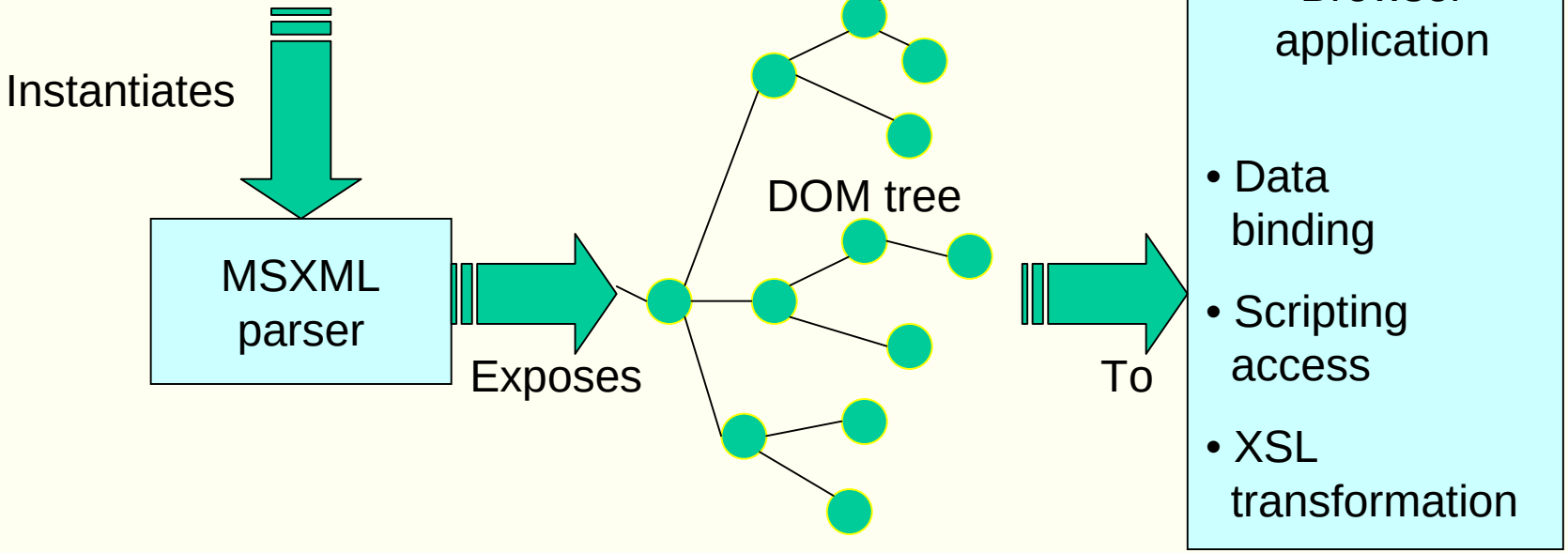
XML viewers

- Microsoft *Internet Explorer 5.5*
  - parser:
    - checks well-formedness, can be forced to check validity
  - CSS:
    - partial support for CSS 1.0, no support for CSS 2.0
  - XSL:
    - built-in MSXML parser: support for "Microsoft XSL" (XSLT working draft)
    - MSXML 3.0 parser: complete support for XSLT standard
  - DOM:
    - built-in MSXML parser: support for DOM 1.0
    - MSXML 3.0 parser: complete support for DOM 2.0
  - proprietary extensions:
    - experimental support for XQL and XML-Data Reduced
    - proprietary support for XML Data Islands in HTML pages accessible through DSO (Data Source Objects) or the DOM

XML Data Island in HTML page

```
<XML id="AuthorInfo">
<author>
    <name>Hans Arents</name>
    <affiliation>I.T. Works</affiliation>
</author>
</XML>
```

XML Data Island



XML viewers

- Netscape *Navigator 6.0* / Mozilla 0.8
  - parser:
    - checks well-formedness only
  - CSS:
    - full support for CSS 1.0, partial support for CSS 2.0
  - XSL:
    - no support, but XSL processor can be plugged in (Mozilla 0.9?)
    - based on open source XSL processor *TransforMiiX*
  - DOM:
    - support for DOM 1.0, partial support for DOM 2.0
  - proprietary extensions:
    - XUL (XML-based User interface Language) for representing the design, features, objects and layout of all browser user interfaces
- Commercial product: *DocZilla* XML/SGML viewer

XML viewers

	Browser			
Feature	**Netscape 6 Preview Release 1**	**Opera 4 beta 3**	**Internet Explorer 5.0/Mac**	**Internet Explorer 5.5/Win**
XML display without style sheet	Everything as inline	Everything as inline	Structured presentation	Structured presentation
Uses W3C PI to connect style sheets	yes	yes	yes	yes
`display:inline`	yes	yes	yes	yes
`display:block`	yes	yes	yes	yes
`display:none`	yes	yes	yes	yes
`display:list` (and `list-item`)	yes	yes (numbering issues)	yes (numbering issues)	no
`display:table` (and contents)	yes	yes (some interactions with embedded HTML)	no	no
XLink support	yes (3/3/98 draft)	through CSS extensions	no	no
Embedded HTML support	yes (using HTML 4.0 URI as namespace)	yes (using HTML 4.0 URI as namespace, some problems with HTML `img` elements)	yes (using `html` prefix, some problems with HTML `a` elements)	yes (using `html` prefix)
XSLT support	no (in development)	no (no announcements)	yes, based on old draft (new processor in development)	yes, based on old draft (new processor in development)
Release status	Preview Release	Beta	Shipping	Preview Release

- Internet Explorer 5.5
  - CSS: 7/10
  - XSLT: 9/10
- Netscape Navigator 6.0
  - CSS: 9/10
  - XSLT: 0/10
- Opera 5.0
  - CSS: 8/10
  - XSLT: 0/10

… so your best bet now is Internet Explorer!

XSLT processors

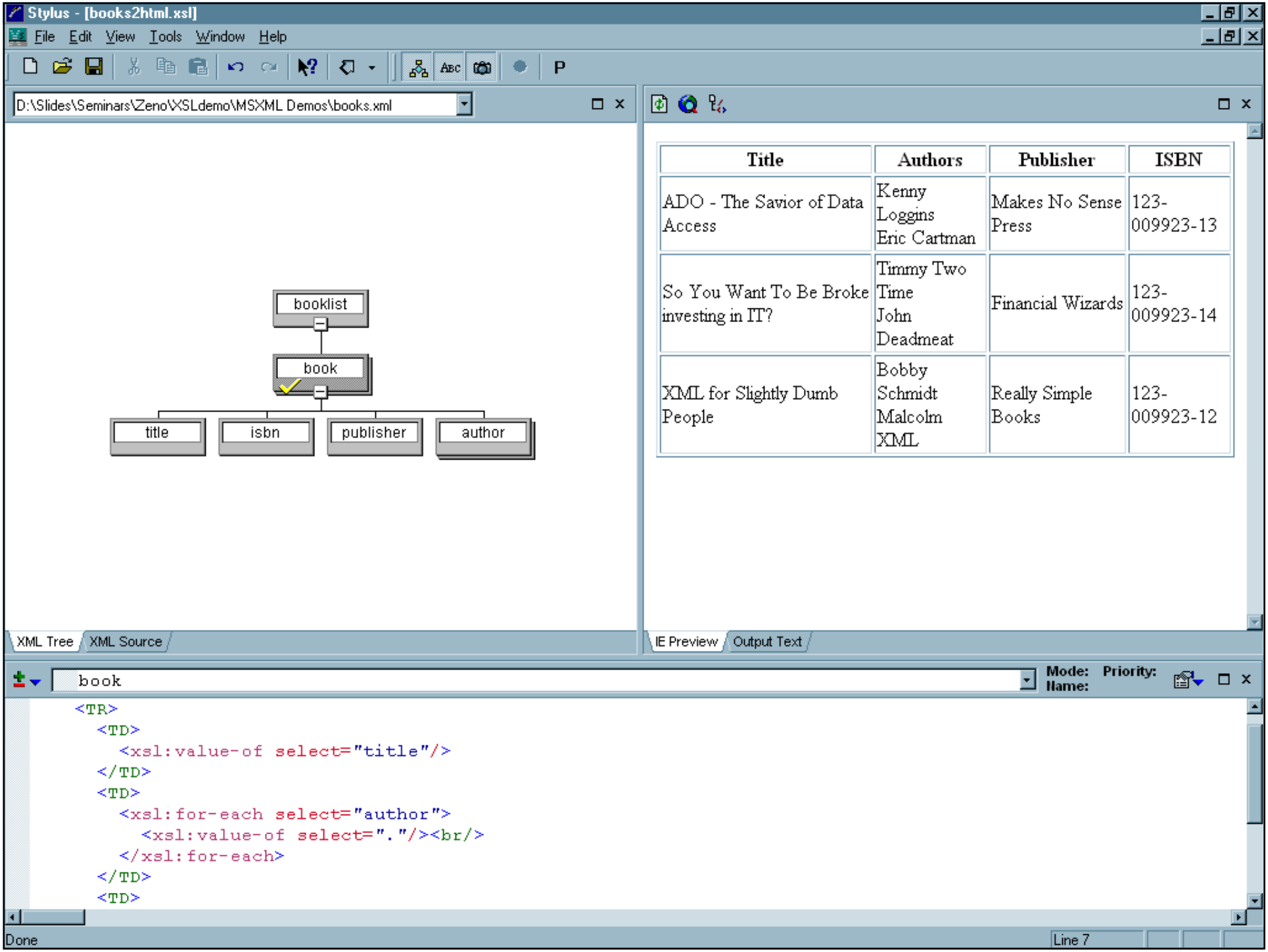
- Windows:
  - Michael Kay's *SAXON* (Java) or *Instant SAXON* (Windows executable)
    http://users.iclway.co.uk/mhkay/saxon/
    - adheres very closely to the official XPath and XSLT standard
    - to be used from the command line, or wrapped in an applet or servlet
    - has extensions for calling Java code and handling multiple documents
  - Infoteria *iXSLT* **http://www.infoteria.com/**
    - (expensive) XSL processor, available as a EXE / DLL / COM
- Java:
  - IBM *LotusXSL* **http://www.alphaworks.ibm.com/**
    - proof-of-technology XSLT processor
    - to be used from the command line, or wrapped in an applet or servlet
    - *XML Enabler*
      - sniffs incoming browser type
      - selects appropriate XSL stylesheet

XSLT processors

- Java:
  - Apache's *Xalan* http://xml.apache.org/xalan-j/
    - part of the Apache XML toolset
    - leading Java-based XSLT processor
    - adheres very closely to the official XPath and XSLT standard
    - to be used from the command line, or wrapped in an applet or servlet, or as a module in your own Java program
    - uses the Apache Xerces XML parser by default, but it can interface to any XML parser that conforms to the DOM level 2 or SAX level 1 specification
    - supports proprietary extensions:
      - growing number of proprietary extension elements and functions
      - Java and scripting language (JavaScript, PerlScript or Python) extensions
    - version 2.0 implements the **Tr**ansformation **A**PI for **X**ML (TrAX), now part of the **J**ava **A**PI for **X**ML **P**rocessing 1.1 (JAXP)
      - standard API to apply XSL transformations to XML documents
      - can use compiled XSLT stylesheets for improved performance

XSL editors

- Windows:
  - eXcelon Corp. *Stylus* http://www.exceloncorp.com/products/
    - tree view of the structure of the XML data
    - syntax-aware stylesheet editing and debugging
    - can use both built-in XSLT processor and MSXML parser
- Java:
  - IBM *XSL Editor* http://www.alphaworks.ibm.com/
    - tree view of the structure of the XML data
    - automatically generates XPath syntax from sample XML data
    - *XSL Trace*
      - steps through XSL stylesheet and shows active transformation rules
    - *Visual XML Transformation*
      - input the source DTD and visually define the target DTD
      - generates XSL stylesheet that transforms source into result XML data


Stylus - [books2html.xsl]
File Edit View Tools Window Help
D:\Slides\Seminars\Zeno\XSLdemo\MSXML Demos\books.xml
booklist
book
title
isbn
publisher
author
XML Tree
XML Source
Title
Authors
Publisher
ISBN
ADO - The Savior of Data Access
Kenny Loggins Eric Cartman
Makes No Sense Press
123-009923-13
So You Want To Be Broke investing in IT?
Timmy Two Time John Deadmeat
Financial Wizards
123-009923-14
XML for Slightly Dumb People
Bobby Schmidt Malcolm XML
Really Simple Books
123-009923-12
IE Preview
Output Text
book
Mode:
Priority:
Name:
<TR>
<TD>
<xsl:value-of select="title"/>
</TD>
<TD>
<xsl:for-each select="author">
<xsl:value-of select="."/>
</xsl:for-each>
</TD>
<TD>
Done
Line 7

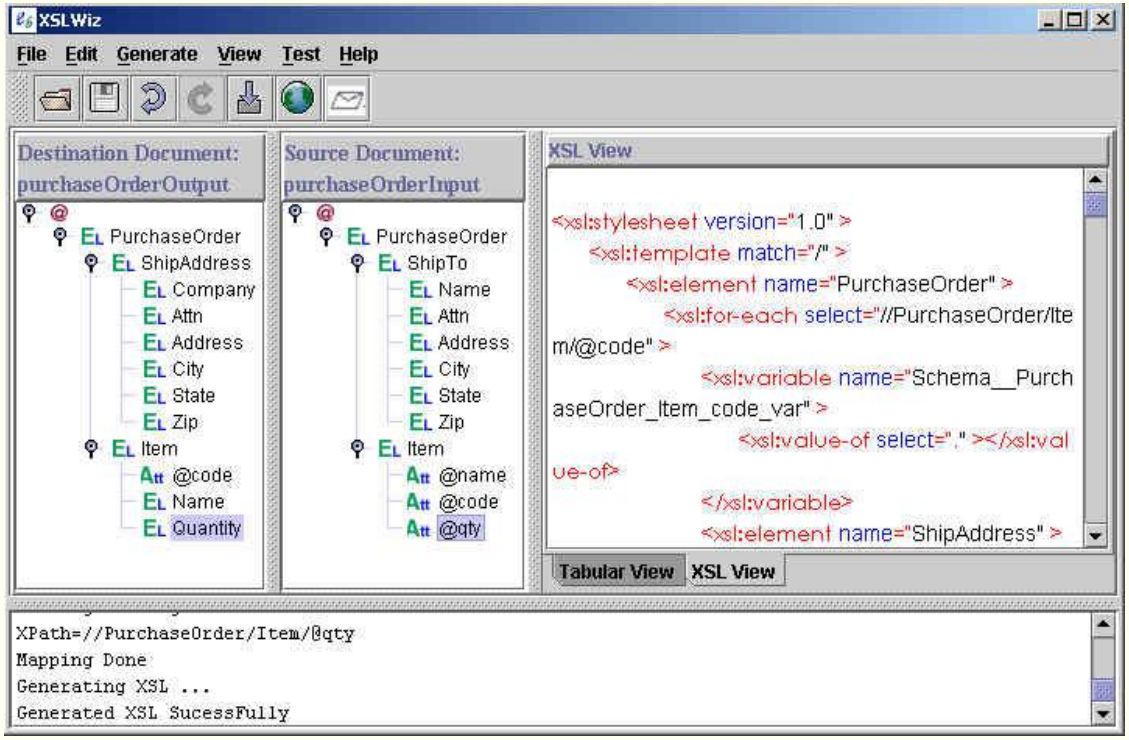
XSL editors

- Windows:
  - IPNovation *XSL Studio* **http://www.ipnovation.com/xslstudio.htm**
    - tool for developers
    - XML validation and XSL transformation using MSXML parser
    - comes with *XSL Developer Guide*, a quick reference guide to XSL syntax
    - previewing using IE, synchronization between XSL code and HTML preview

- Windows:
  - Whitehill *<XSL> Composer* **http://www.whitehill.com/**
    - tool for non-developers
    - WYSIWYG development environment
    - drag and drop XML elements onto an HTML preview pane
    - dynamically creates XSL stylesheets to generate HTML (+ CSS)

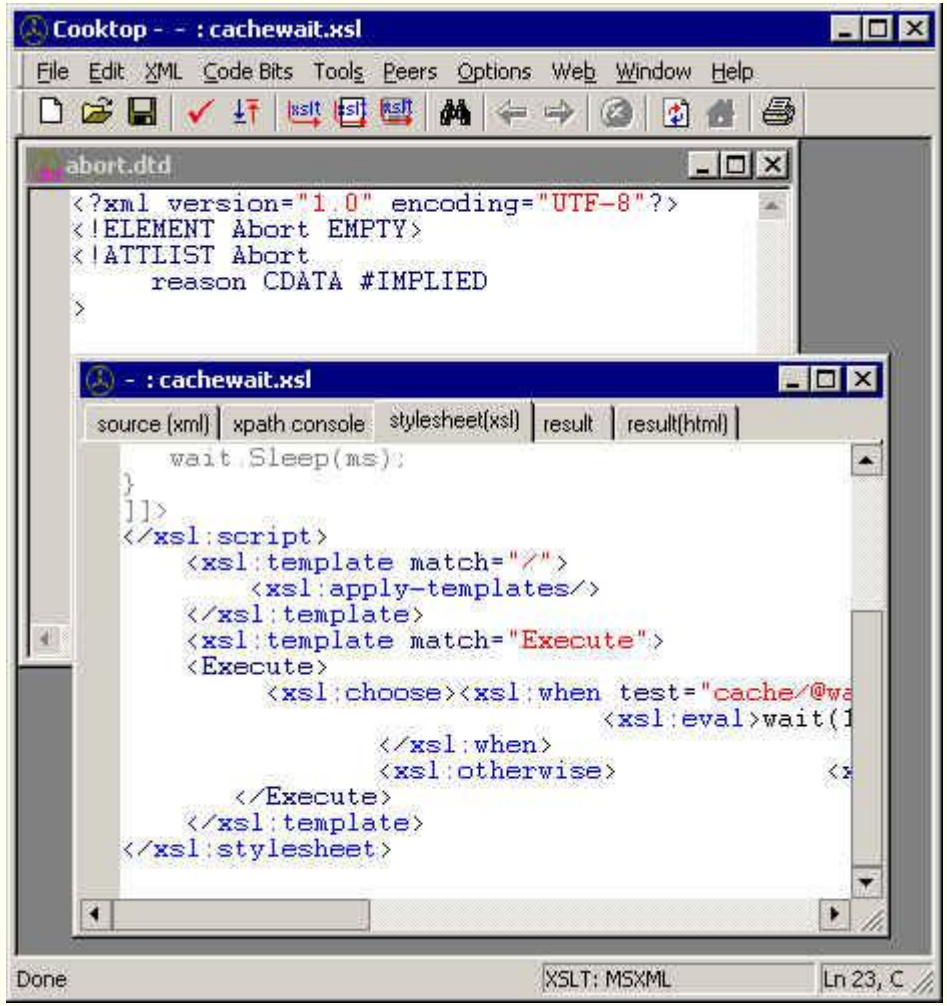
XSL generators

- EBProvider *XSLWiz*
  http://www.ebprovider.com/products/xslwiz.html
  - visually map XML documents conforming to one XML Schema to another XML Schema and generate the corresponding XSL stylesheet
  - infer an XML Schema from a sample XML document
  - convert a DTD to an XML Schema



XML Integrated Development Environments

- XMLeverywhere *XML Cooktop*
  http://xmleverywhere.com/cooktop
  - color-coded XML editing, MS Word spell checker, XML code bits library
  - XSL stylesheet development & testing, XHTML conversion
  - has a document navigator with support for XPath creation and testing
  - uses MSXML parser for XML checking/validation and XSL transformations
  - uses IE for previewing HTML
  - it's FREE!

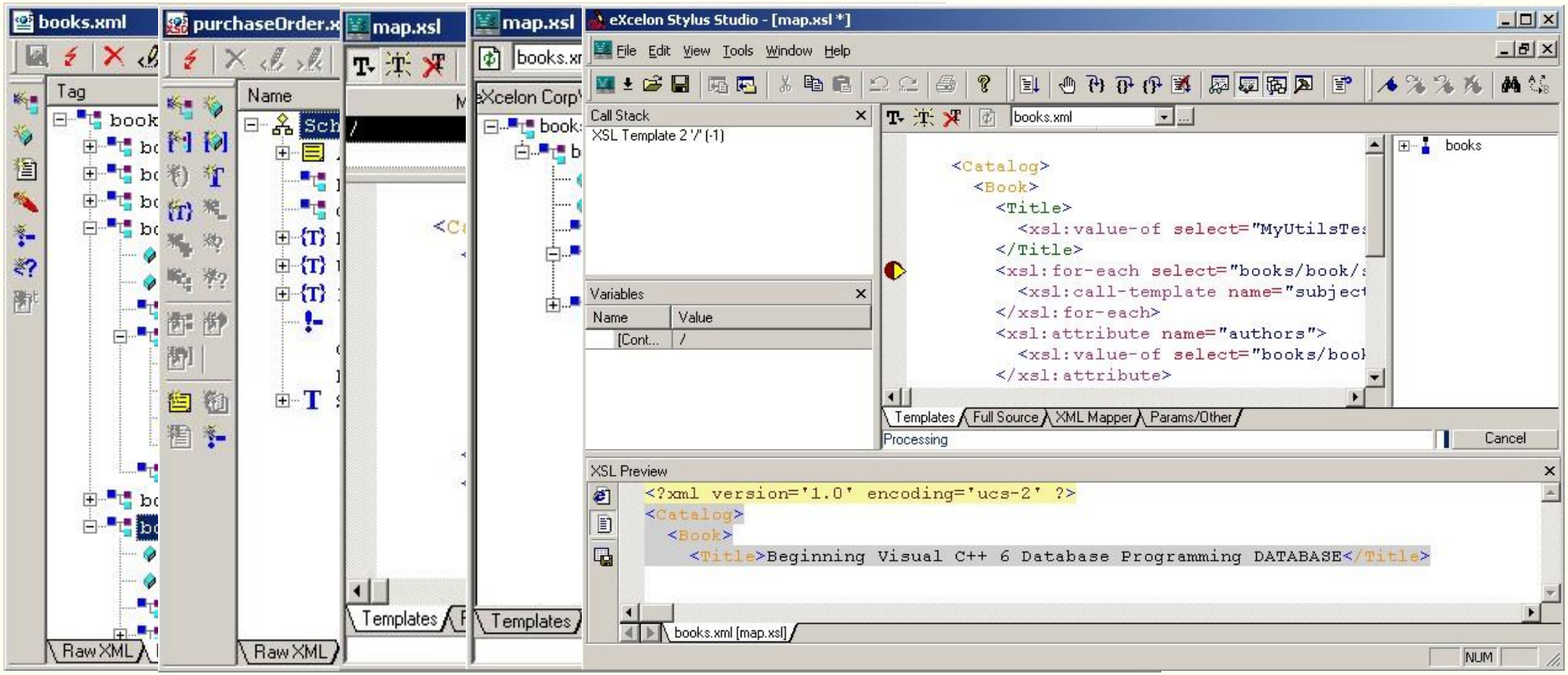


XML Integrated Development Environments

- eXcelon Corp. *Stylus Studio*
  http://www.exceloncorp.com/products/
  - project management
    - manage related files in project views
  - visual XML editor
    - syntax highlighting, Sense:X tag completion, tree and grid views
    - auto DTD generation, XPath support: XPath query builder / result mapper
  - visual XML schema editor
    - supports DTD and XML Schemas
  - visual XSLT editor
    - syntax highlighting, Sense:X tag completion, support for templates
    - template repository, support for Java extension functions
  - visual XML-to-XML mapper
    - creates XSLT stylesheets that transform between different XML schemas
    - supports DTD and XML Schemas

XML Integrated Development Environments

- Java source editor
  - edit and compile Java extension functions
- integrated Java and XSLT debugger
  - set breakpoints in Java extension functions and XSLT sources
  - step through XSLT into Java and back into XSLT



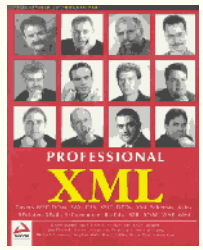
XSL developer sites

- The *XSLT* site
  - **http://www.xslt.com/**
- The *XSLinfo* site
  - **http://www.xslinfo.com/**
- The *XSL FAQ*
  - **http://www.dpawson.co.uk/xsl/xslfaq.html**
- The *XSLT Quick Reference*
  - **http://www.mulberrytech.com/quickref/**
- Microsoft's *XSL Developer's Guide*
  - **http://msdn.microsoft.com/xml/XSLGuide/**
- Web Developer's Virtual Library *XSL section*
  - **http://wdvl.com/Authoring/Languages/XSL/**
- The *XSL-List* developers discussion forum
  - **http://www.mulberrytech.com/xsl/xsl-list/**

XSL developer reading material

- *XSLT Programmer's Reference*
  by Michael Kay, David Sussman
  - **ISBN:1861003129**

- *Professional XML* (chapter 9)
  by Mark Birbeck et al.
  - **ISBN:1861003110**



- *XML Bible* (chapter 6)
  by Elliotte Rusty Harold
  - **ISBN:0764532367**

- *Practical Transformation Using XSLT and XPath*
  by Ken Holman
  - **ISBN:1894049047**
  - **http://www.CraneSoftwrights.com/**