Title

Thorsten S. Daum

UpstateConsultants.com

Introduction to XML

http://upstateconsultants.com/uc/docs

What is XML anyway?

- XML = Data + Descriptions (Markup) + APIs

```
<?xml version="1.0" encoding="UTF-8"
      standalone="yes"?>
<address format="US">
  <name type="individual">John Doe</name>
  <street>123 Elm St</street>
  <city>Anytown</city>
  <region type="state">New York</region>
  <postal-code>12345</postal-code>
  <country>U.S.A.</country>
</address>
```

What is XML anyway? *(cont'd)*

- "Data + Markup" around since 1970s: SGML
  - Never caught on, except for niche markets, e.g., dictionary publishing
  - So how is XML new?

How is XML new?

- "Data + Markup" around since 1970s: SGML
  - Never caught on, except for niche markets, e.g., dictionary publishing
  - So how is XML new?
- Standardized APIs are new in XML
  - Applications use available code to read and write XML
    - Makes it easy to develop and use Web services

Properties of XML

- Strictly hierarchical
  - ➢ Always one and only one root element
    - No more `<head>...</head><body>...</body>`
- Must be well-formed
  - ➢ Every start element must have an end element
    - No more `text<p><p><p><p>text after space`
  - ➢ Proper nesting
    - `<i><b>Important!</i></b>` will never process!

Well-formed XML

- New notation for empty elements
  - `<hr/>` or `<hr />` *(for compatibility with HTML)*
  - Can have attributes, e.g., `<hr class="blue"/>`
- Strict syntax for attributes
  - Attribute values must be enclosed in quotes *(' or ")*
    - `<foo key='1' val="2"/>` is valid, albeit bad style
  - No more attribute minimization
    - Always `<option checked="yes">`
    - Never `<option checked>`

XML Document – Components

- Every document starts with a "Prolog"
  - XML declaration
  - Document Type Declaration(s)
- Root Element
  - Character Data
  - Markup
- Procession Instructions
- Comments `<!-- Just like in HTML -->`

Markup

- Start and end elements, empty elements
- CDATA *(unparsed)*
- Entity References and Character References
- Procession Instructions
- Comments `<!-- Just like in HTML -->`

XML Declaration

- *<u>Must</u>* come first in an start with an XML document

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

(minimal)

```
<?xml version="1.0"
      encoding="UTF-8"
      standalone="yes"?>
```

Default: "UTF-8" or "UTF-16"

Default: "no"

"yes" means there can be no external DTD

Well-Formed vs. Valid Documents

- All documents must be well-formed
  - Syntax rules
  - If it's not well-formed, it's not XML
    - Parsers *must* abort processing
- Validity: additional constraint
  - Optional, can be turned on and off
    - Validating/Non-validating parsers
  - Semantics
    - Defined in DTD or XML Schema

Document Type Declaration

- Series of markup declarations that provide a grammar for a class of documents
- Can be implemented as
  - External subset,
  - Internal subset,
  - Or both *(where internal subset overrides external subset.)*
- Together, they form the Document Type *Definition* (DTD),
  - N.b., no acronym for Document Type *Declaration*!

External Subset: Declaration

```
<?xml version="1.0" ?>
<!DOCTYPE address SYSTEM "address.dtd">
<address format="US">
  <name type="individual">John Doe</name>
  <street>123 Elm St</street>
  <city>Anytown</city>
  <region type="state">New York</region>
  <postal-code>12345</postal-code>
  <country>U.S.A.</country>
</address>
```

address.dtd

```
<!ELEMENT address (name, street+, city,
                   region?, postal-code?,
                   country)>
<!ATTLIST address format CDATA "US">
<!ELEMENT name (#PCDATA)>
<!ATTLIST name
          type (individual, org) "individual">
<!ELEMENT street (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT region (#PCDATA)>
<!ATTLIST region type CDATA #IMPLIED>
<!ELEMENT postal-code (#PCDATA)>
<!ELEMENT country (#PCDATA)>
```

External and Internal Subsets

```
<?xml version="1.0" ?>
<!DOCTYPE address SYSTEM "address.dtd">
<!DOCTYPE address [
  <!ATTLIST region type #FIXED "US">
]>
<address format="US">
  <name type="individual">John Doe</name>
  <street>123 Elm St</street>
  <city>Anytown</city>
  <region type="state">New York</region>
  <postal-code>12345</postal-code>
  <country>U.S.A.</country>
</address>
```

Processing Instruction (PI)

- Allow a document to contain instructions to applications
- Will be passed through to processing application
- Not part of the document's character data
- E.g., Mozilla can be caused to perform an XSL Transformation on an XML document with an PI such as

```
<?xml-stylesheet href="address-book.xsl"
                 type="text/xsl"?>
```

Text

- Historically known as "PCDATA"
- A sequence of characters
  - Markup
  - Character data
- Characters
  - tab, carriage return, line feed
  - Any legal Unicode or ISO/IEC 10646 character
- Character data
  - Any text that is not markup

CDATA Sections

- Can occur anywhere character data can occur
- Escape blocks of text that would be markup
  - Useful for showing XML "source" code
- Start with "`<![CDATA[`"
- End with "`]]>`"

Not parsed as markup

```
<element-doc>
  Example: <![CDATA[<country>U.S.A.</country>]]>
</element-doc>
```

API's

- Standardized interfaces to process XML
  - Process XML document as object tree: DOM
  - Process XML document sequentially: SAX
- What makes XML new and exciting
- Implemented in wide variety of languages
  - Java (`javax.xml`, Apache XML project)
  - C++, Perl, PHP, proprietary Microsoft languages, ...

Document Object Model (DOM)

- Entire document is processed and converted into a tree
  - Elements are nodes in the tree
  - Methods to access and manipulate tree nodes
- Pros
  - Access to entire document
  - Reorder elements (nodes)
- Cons
  - Large documents can be unmanageable

Simple API for XML (SAX)

- Documents are processed sequentially
  - Methods are called for each start/end element and text
- Pros
  - Process huge (and even streamed) documents
  - Create XML by calling methods
  - Fast
- Cons
  - Document isn't persistent, hard to reorder elements

Recommended Reading

- The only authoritative resource: **`http://www.w3.org/`**
  - XML Specification *("W3C Recommendation")* **`http://www.w3.org/TR/REC-xml/`**
  - DOM: **`http://www.w3.org/DOM/`**
  - SAX: **`http://www.saxproject.org/`**
- Seasoned IT Professionals: Skip the XML books!
- Not covered today: Namespaces, XML Schema