

# **XML and XSLT Overview**

## **XML and XSLT Overview**

Instructor: Rick Palmer, SCWCD

[rick@online-ettraining.com](mailto:rick@online-ettraining.com)

# Topics Covered

- XML
- XSLT

# XML Introduction

- ❑ XML = eXtensible Markup Language
  - ⇒ Released Feb 1998 by W3C (World Wide Web Consortium)
- ❑ Designed to separate “data” from “formatting”.
- ❑ HTML defines both data and how a web browser should display the data (formatting):

```
<html>
  <body><table>
    <tr><td><b>Developer: </b> </td><td><i>Bill Gates</i></td></tr>
  </table></body>
</html>
```

- ❑ XML only defines data (notice there are no HTML formatting instructions like <table>, <td>, <b>, or <i>):

```
<employee>
  <developer name="Bill Gates" />
</employee>
```

# XML Has Many Uses

- ❑ Deployment descriptors (configuration files)
- ❑ Transporting hierarchical data on the web (Ajax)
- ❑ Platform-independent web services (SOAP)
- ❑ Documentation that will be displayed in multiple formats (e.g. HTML, Help files, and PDF)
- ❑ Integration with legacy data (XSLT)
- ❑ Error logging
- ❑ Etc... Etc... Etc...

# XML Rules

## ❑ XML must be “Well-formed”

⇒ Single top-level root element (`<web-app></web-app>`)

⇒ Closing tags required, unlike HTML

⇒ No overlapping tags; nested is ok though:

```
<employee><name>Jeff</name></employee>
```

⇒ Case sensitive (`<Name>` is different than `<name>`)

⇒ No invalid characters (`<`, `>`, `&`)

⇒ No white space in element names

## ❑ XML *can* be “Valid”, but it’s not required

⇒ XML is valid if it conforms to a DTD or XML Schema.

❑ A DTD or XML Schema defines the XML elements and attributes that should be contained in the XML document (not covered in this course)

# Namespaces

- ❑ Some XML documents contain similarly-named tag names (e.g. employees from two companies)
- ❑ Namespaces uniquely identify similarly-named tags
- ❑ Prefix + URI = unique tag name

```
<?xml version="1.0" encoding="UTF-8"?>
<employees xmlns:isi="http://www.integrated_services_inc.com/employee" xmlns:ms="http://www.microsoft.com/employee" >
  <isi:employee department="Dev">
    <isi:name>Skip Chapman</isi:name>
    <isi:income>$90,000</isi:income>
  </isi:employee>
  <ms:employee department="Dev">
    <ms:name>Doug Ashley</ms:name>
    <ms:income>$60,000</ms:income>
  </ms:employee>
</employees>
```

- ❑ Both elements and attributes can be qualified by a namespace identifier.
- ❑ Namespaces are optional, but are recommended to ensure element uniqueness inside xml documents.

# XSLT Introduction

- ❑ XSLT: Extensible Stylesheet Language for Transformations
  - ⇒ XSLT converts XML to other document formats
- ❑ Converts XML

```
<?xml version="1.0" encoding="UTF-8"?>
<company>
  <employees>
    <employee department="Dev">
      <name>Bill Gates</name>
      <income>$670,000</income>
    </employee>
  </employees>
</company>
```

- ❑ To HTML, XML, CSV, PDF, etc..

```
<p>Dear Bill Gates,</p>
<p>We regret to inform you that the Dev department
  can no longer afford your $670,000 services.</p>
<hr width="100%" />
```

# XSLT Building Blocks

- ❑ Root stylesheet element with namespace designator

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

- ❑ Output type element

```
<xsl:output method="html" /> (could also use "xml", "text", etc)
```

- ❑ XSLT template elements

```
<xsl:template match="/" > <!-- Process the document root -->
  <html>
    <body>
      <!-- Process all employee elements -->
      <xsl:apply-templates select="//employee" />
    </body>
  </html>
</xsl:template>
```

# XSLT Building Blocks (cont.)

## □ Applying XSLT templates recursively

```
<xsl:template match="//employee" <!-- Process all employee elements -->
  <p>Dear <xsl:value-of select="name" />, </p>
  <p>We regret to inform you that the
    <xsl:value-of select="@department"/> department
    can no longer afford your <xsl:value-of select="income"/>
    services.</p>
  <hr width="100%" />
</xsl:template>
```

## □ Selecting the value of an XML element

```
<xsl:value-of select="name" />
```

## □ Selecting the value of an XML attribute

```
<xsl:value-of select="@department" />
```

# XSLT Transformation with IE

- ❑ Internet Explorer includes its own XSLT processing engine.
- ❑ Include an `xml-stylesheet` directive at the top of the XML document that points to the stylesheet:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="layoffs.xslt"?>
<company>
  <employees>
    <employee department="Dev">
      <name>Bill Gates</name>
      <income>$670,000</income>
    </employee>
  </employees>
</company>
```

- ❑ Open the xml file in the browser to transform it to HTML

# Resources

❑ Java Web Services Tutorial:

<http://java.sun.com/webservices/docs/1.6/tutorial/doc/>

❑ Apache XML Project: <http://xml.apache.org/>

❑ Java Guru: <http://www.jguru.com/>

❑ Java World: <http://www.javaworld.com/>

❑ MSDN XSLT Reference:

<http://msdn.microsoft.com/library/en-us/xmlsdk/html/xmrefxsltrefer>