Chapter 5
Data Access - XML

Mrs. Swati Satpute
Fergusson College

Objective

Understanding

- What is a XML.
- How to manipulate XML files.

What is XML?

- XML stands for E**x**tensible **M**arkup **L**anguage.
- XML is way of storing data in text format in hierarchical representation.
- Human readable, can also be read by any computer.
- XML can be user defined.
- XMLs are perfect format for transferring data over the net.

What is XML?

- XMLs are case sensitive.
- XML is not a language instead it’s a standard for defining languages.
- Best part of working with XML is one can define his own vocabulary.
- Restriction on vocabulary is possible by enforcing schema.

XML Parsers

- Parsers are the XML program which reads the XML and analyze them on individual element level.
- Parsers can reject document in case of illegal xml.
- Known Parsers
  - SAX (Simple API for XML)
  - DOM (Document Object Model)

XMLDocument

- Can be referred as container of the XML data.
- XMLDocument can be XML file on NTFS or it can represent XML in memory stream.
- XMLDocument may contain following:
  - XMLElements
  - Attributes

XMLElements

- XMLElement is the node which actually contain data.
- XMLElement can be represented as <Student>.

```
<Student>
        <Name> John Baker </Name>
        <RollNo> 200</RollNo>
</Student>
```

- Its possible to have “Empty” elements.
  <Address></Address>
- Shorthand notation
  <Address/>

XMLElements

Rules

- Element name should be enclosed within <>
- Overlapping of elements are not allowed.
- All sub-elements should be closed before closing parent element.

Attributes

- Attributes can be specified within opening tag of an element.

  <Student CourseType=‘MCS’>
- Attribute must be enclosed within single or double quotes.

XML Declaration

- The XML declaration is a processing instruction that identifies the document as being XML.
- All XML documents should begin with an XML declaration.
- Declaration node must be first one in the document.

```
<?xml version="version_number"
   encoding="encoding_declaration"
   standalone="standalone_status" ?>
```

```
<?xml version="1.0" encoding="UTF-16" standalone='yes'>
```

XML Document Structure

- Data is **structured hierarchically**.
- **Single root element**, within it all elements and text nodes are contained.
- Along with root element, XML declaration node can be at the top level.
- No predefined structure under root element.

```
<?xml version="1.0" encoding="UTF-16" standalone='yes'>
<Student>
        <Name> John Baker </Name>
        <RollNo> 200</RollNo>
</Student>
```

XML Namespace

- To avoid the naming conflicts of the XML elements, namespace is used.

- The namespace is defined by the **xmlns attribute** in the start tag of an element.

- The namespace declaration has the following syntax.

  xmlns:*prefix*="*URI*".

XML Namespace

```
<root>

    <h:table xmlns:h="http://www.w3.org/TR/html4/">
    <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
    </h:tr>
    </h:table>

    <f:table xmlns:f="http://www.w3schools.com/furniture">
    <f:name>African Coffee Table</f:name>
    <f:width>80</f:width>
    <f:length>120</f:length>
    </f:table>
</root>
```

Norms for Well-Formed XML

To be a well-formed xml, document should follows following rules.

- XML documents must have a single root element.
- XML elements must have a closing tag.
- XML tags are case sensitive.
- XML elements must be properly nested.
- XML attribute values must be quoted.

Valid XMLs

- A "Valid" XML document is a "Well Formed" XML document, which also conforms to the rules of a Document Type Definition (DTD)

Validating XMLS

- XML Elements and attributes can be restricted by 2 ways
  - DTDs
  - Schemas

DTDs

- The purpose of a DTD (Document Type Definition) is to **define the legal building blocks** of an XML document.
- A DTD defines the document structure with a list of legal elements and attributes.
- DTDs do not allow to specify data types.

Sample - DTD

```
<!DOCTYPE NEWSPAPER [

    < !ELEMENT NEWSPAPER (ARTICLE+)>
    < !ELEMENT ARTICLE (HEADLINE,BYLINE,LEAD,BODY,NOTES)>
    < !ELEMENT HEADLINE (#PCDATA)>
    < !ELEMENT BYLINE (#PCDATA)>
    < !ELEMENT LEAD (#PCDATA)>
    < !ELEMENT BODY (#PCDATA)>
    < !ELEMENT NOTES (#PCDATA)>

    < !ATTLIST ARTICLE AUTHOR CDATA #REQUIRED>
    < !ATTLIST ARTICLE EDITOR CDATA #IMPLIED>
    < !ATTLIST ARTICLE DATE CDATA #IMPLIED>
    < !ATTLIST ARTICLE EDITION CDATA #IMPLIED>
    ]>
```

Validating XMLS

Schemas

- XML-based alternative to DTD.
- An XML Schema describes the structure of an XML document.
- Two forms of Schemas
  - XML Schema definition language (XSD)
    - Open Standard
    - Recommended by W3C
  - XML- Data reduced schemas (XDR)
    - Old standard, Microsoft proprietary.
    - Not recognized by non Microsoft parsers

Sample - XSD

```
<?xml version="1.0"?>
    < schema xmlns:xs ="http://www.w3.org/2001/XMLSchema">
    <xs:element name="StudentCollection">
  <xs:complexType>
   <xs:choice maxOccurs="unbounded">
    <xs:element name="Student">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="Name" type ="xs:string"></xs:element>
       <xs:element name="RollNo" type ="xs:int"></xs:element>
      </xs:sequence>
      <xs:attribute name="StudentType"></xs:attribute>
     </xs:complexType>
    </xs:element>
   </xs:choice>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

XSDs

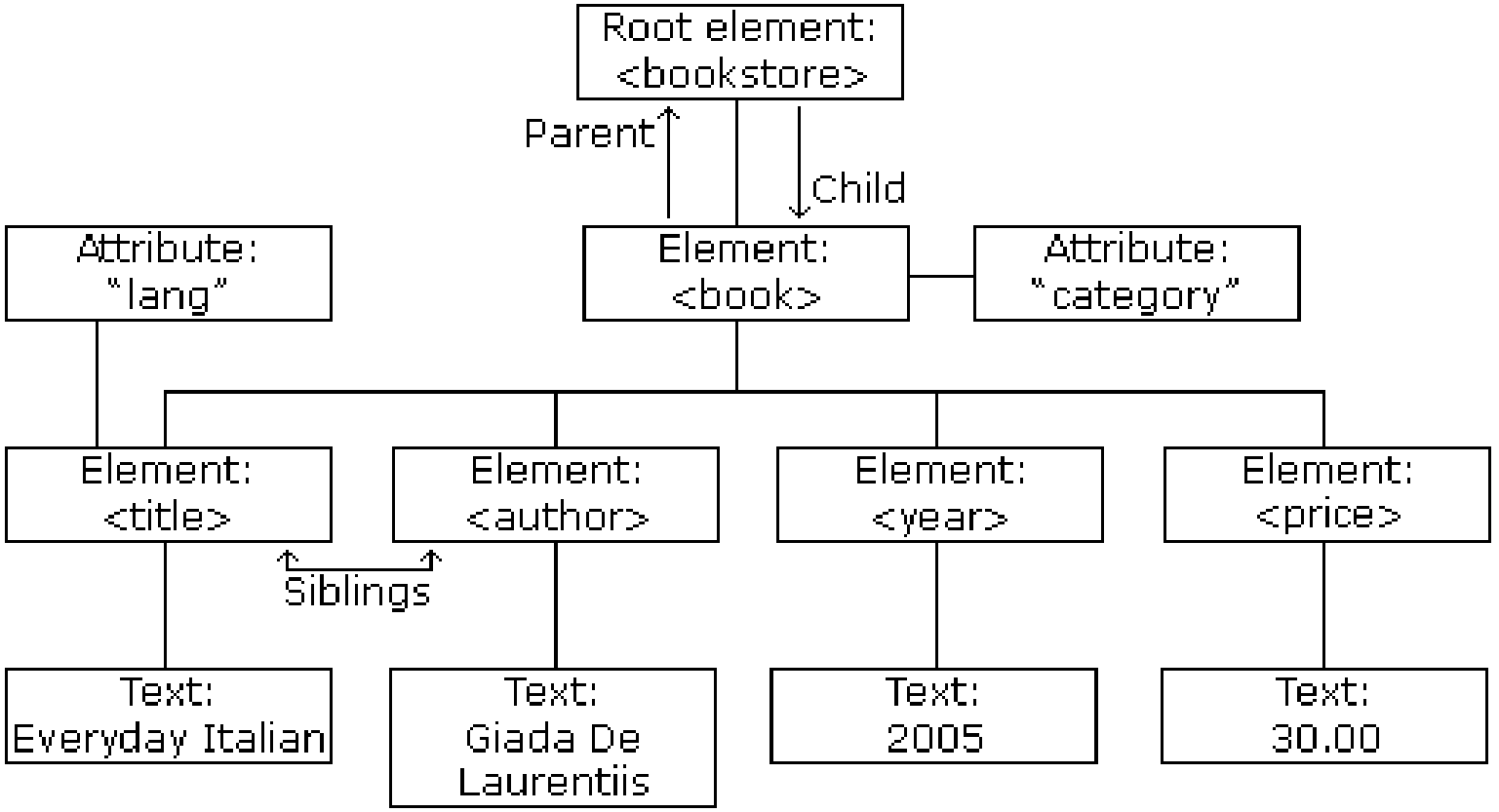
- Specify namespace in schema informs parser that the document contain XML schema definition.

- Each element occurring in XML must have respective element node in schema.

XML Document Object Model

- XML DOM offers multiple classes to access and manipulate XMLs.
- DOM presents an XML document as a tree-structure.
- System.Xml

Sample Tree structure



DOM Classes

Class	Description
XmlNode	Represent single node in the document tree.
XmlDocument	Extends XmlNode class. This class is used to load and save XML data.
XmlElement	Represents single element inXML document. Derived from XmlLinkNode which is derived from XmlNode.
XmlAttribute	Represents single attribute. Derived from XmlNode.
XmlText	Represents text between a starting tag and ending tag
XmlComment	Special type of node. Used to provide information to document reader.
XmlNodeList	Represents collection of nodes.

XmlDocument Class

- XmlDocument holds in-memory representation of XML file.
- XmlDocument class is responsible for maintaining document structure.
- Using this class, one can create, delete modify xml contents.

```
XmlDocument myDoc = new XmlDocument();
myDoc.Load(@"E:\Data.xml");
```

XmlElement Class

- Using document.DocumentElement retrieves root element of the document.
- Using this element, whole document tree can be accessed.

Property	Description
FirstChild	Retrieves the first child of this node.
LastChild	Retrieves the last child of this node.
ParentNode	Returns parent of the current node.
NextSibling	Returns the next node that has same parent
HasChildNodes	Checks whether current node has any child nodes or not

Getting value of node

Property	Description
InnerText	Gets the text of all child nodes and returns a single concatenated string. This means the node names within <> will be ignored.
InnerXml	Gets the text of all Child nodes including tags. InnerXml can be used to insert an xml in other xml.
Value	Only XmlText, XmlComment, XmlAttribute will return the expected value of the node. Rest all nodes will return null.

Inserting nodes

To insert a node, any of the following methods can be used.

Method	Description
AppendChild	Appends a child node to the current node. Newly appended node appears in the bottom of the child nodes.
InsertAfter	Node can be inserted after a particular node. Node location will be decided with respect to the indicator node.
InsertBefore	Node can be inserted before a particular node. Node location will be decided with respect to the indicator node.

Creating Nodes

Method	Description
CreateNode	Create any kind of node.
CreateElement	Creates nodes of XmlElement type.
CreateAttribute	Creates node of XmlAttributes type.
CreateTextNode	Creates node of type XmlTextNode
CreateComment	Creates Comment node.

Selecting Nodes

Method	Description
SelectSingleNode()	Selects a single node. XPath query can be formed to fetch the node. If query returns more than one nodes, then first node will be returned.
SelecNodes()	Returns nodes collection in the form of XmlNodeList class.

XPath

- It’s a query language for XML Documents.

Purpose	XPath Query samples
Select current node	.
Select parent of current node	..
Selecting all child nodes of the current node	*
Selecting all nodes with specific name	Student
Selecting attribute of current node	@StudentType
Selecting all attributes of current type	@*

Xpath Queries

Purpose	XPath Query samples
Selecting child node by index	element[2]
Selecting all text nodes of the current node	text()
Selecting all nodes with specific name	//Student
Selecting all nodes with particular parent and child	//Student/Name
Selecting a node satisfying value criterion	//Student[name='user1']
Selecting a node satisfying attribute value criterion	//Student[@StudentType='current']

Summary

References

- Book referred “Beginning Visual C# 2010” by Wrox publication.
- http://www.w3schools.com/xml/xml_namespaces.asp

Question Bank

1. What is XML parser? Name different XML parsers.
2. What are the norms of well-formed XML?
3. What does valid XML means?
4. What are different xmls validating techniques?
5. What is XmlDocument class? Explain.
6. What is Xpath?