XML Basics

Two Patterns of XML

- Element Centric
- Attribute Centric

Element Centric Example

```
<person>
    <firstname>John</firstname>
    <lastname>Doe</lastname>
</person>
```

Attribute Centric Example

<person firstname="John"
lastname=“Doe" />

Element Centric

- Bigger size
- Easy to describe complex type
- Easy to describe Nullable data
- Faster to parse
- Works better with WCF framework
- More readable by most parsers

Attribute Centric

- Not very interoperable, since most XML parsers think data is presented by Element, Attributes are used to describe the element
- Difficult to present nullable value for some data types
- Difficult to express complex types
- Attributes cannot be duplicated

Attribute Centric (continued)

- Smaller size
- More readable for human
- Attributes are not order sensitive

Element Centric vs Attribute Centric

- Can have a mix of both
- Better be consistent
- Most of the time, these two patterns are equivalent

XML Validation

- XML with correct syntax is "Well Formed"
- XML validated against a DTD (Document Type Definition) is "Valid"
- DTD defines structure of an XML document
- W3C supports an XML-based alternative to DTD, called XML Schema

XML Validation (continued)

- "Valid" XML document is "Well Formed", also conforms to a DTD
- Typed XML has associated schema
- Untyped XML does not have associated Schema

XML Syntax Rules

- Documents must have a root element
- Elements must have a closing tag
- Tags are case sensitive
- Elements must be properly nested
- Attribute values must be quoted

Well Formed XML Document Example

```
< ?xml version="1.0" encoding="ISO-8859-1"?>
< note>
    < to>John</to>
    < from>Peter</from>
    < heading>Reminder</heading>
    < body>Don't forget the meeting!</body>
< /note>
```

Valid XML Document Example

```
< ?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE note SYSTEM "Note.dtd">
< note>
    < to>John</to>
    < from>Peter</from>
    < heading>Reminder</heading>
    < body>Don't forget the meeting!</body>
< /note>
```

JSON
(JavaScript Object Notation)

- Open standard for storing and exchanging text info
- Smaller than XML
- Faster and easier to parse
- Derived from JavaScript but language-independent
- With parsers available for many languages
- Primarily used to transmit data between server and web application
- Alternative to XML