

XML

XML is a software- and hardware-independent tool for storing and transporting data.

What is XML?

XML stands for eXtensible Markup Language

XML is a markup language much like HTML

XML was designed to store and transport data

XML was designed to be self-descriptive

XML is a W3C Recommendation

The Difference Between XML and HTML

XML was designed to carry data - with focus on what data is

HTML was designed to display data - with focus on how data looks

XML tags are not predefined like HTML tags are

XML Separates Data from Presentation

XML does not carry any information about how to be displayed.

The same XML data can be used in many different presentation scenarios.

XML Document

Books.xml

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

```
<bookstore>
```

```
  <book category="cooking">
```

```
    <title lang="en">Everyday Italian</title>
```

```
    <author>Giada De Laurentiis</author>
```

```
    <year>2005</year>
```

```
    <price>30.00</price>
```

```
  </book>
```

```
  <book category="children">
```

```
    <title lang="en">Harry Potter</title>
```

```
    <author>J K. Rowling</author>
```

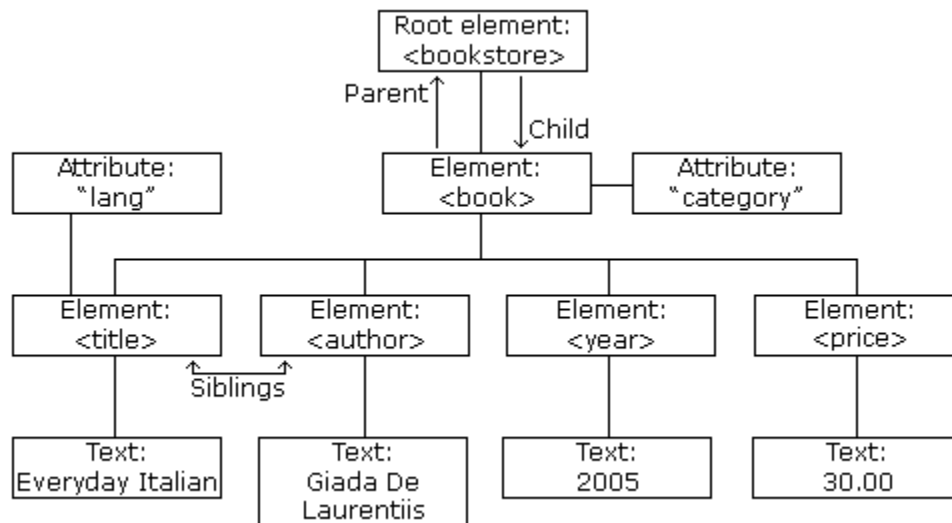
```
    <year>2005</year>
```

```

    <price>29.99</price>
  </book>
<book category="web">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>

```

XML Tree Structure



XML Namespaces

XML Namespaces provide a method to avoid element name conflicts.

Name conflicts in XML can easily be avoided using a name prefix.

This XML carries information about an HTML table, and a piece of furniture:

```

<h:table>
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>

```

```
<f:table>
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

XSL

Before learning XSLT, we should first understand XSL which stands for EXtensible Stylesheet Language. It is similar to XML as CSS is to HTML.

Need for XSL

In case of HTML document, tags are predefined such as table, div, and span; and the browser knows how to add style to them and display those using CSS styles. But in case of XML documents, tags are not predefined. In order to understand and style an XML document, World Wide Web Consortium (W3C) developed XSL which can act as XML based Stylesheet Language. An XSL document specifies how a browser should render an XML document.

Following are the main parts of XSL –

- **XSLT** – used to transform XML document into various other types of document.
- **XPath** – used to navigate XML document.
- **XSL-FO** – used to format XML document.

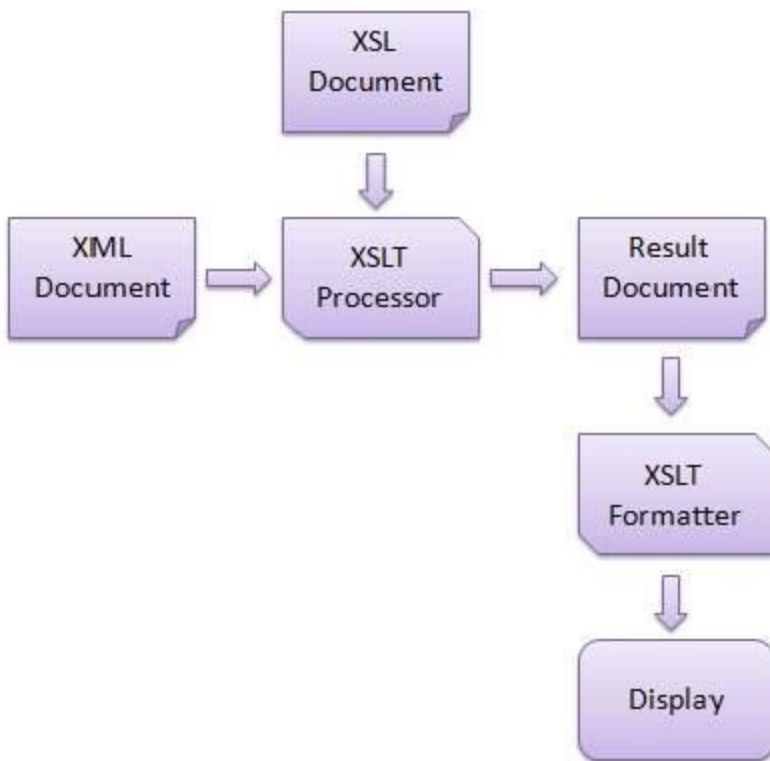
XSLT

What is XSLT

XSLT, Extensible Stylesheet Language Transformations, provides the ability to transform XML data from one format to another automatically.

How XSLT Works

An XSLT stylesheet is used to define the transformation rules to be applied on the target XML document. XSLT stylesheet is written in XML format. XSLT Processor takes the XSLT stylesheet and applies the transformation rules on the target XML document and then it generates a formatted document in the form of XML, HTML, or text format. This formatted document is then utilized by XSLT formatter to generate the actual output which is to be displayed to the end-user.



Advantages

Here are the advantages of using XSLT –

- Independent of programming. Transformations are written in a separate xsl file which is again an XML document.
- Output can be altered by simply modifying the transformations in xsl file. No need to change any code. So Web designers can edit the stylesheet and can see the change in the output quickly.

Example

Book.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="book.xsl"?>
<catalog>
  <book>
    <isbn>1001</isbn>
    <title>programming in c</title>
    <author>shiva</author>
```

```
        <price>78</price>
    </book>
<book>
    <isbn>1002</isbn>
    <title>java</title>
    <author>ravi</author>
    <price>78</price>
</book>
</catalog>
```

Book.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html><body>
<table border="2">
<xsl:for-each select="catalog/book">
    <tr>
        <td> <xsl:value-of select="isbn"/> </td>
        <td> <xsl:value-of select="author"/> </td>
        <td> <xsl:value-of select="title"/> </td>
        <td> <xsl:value-of select="price"/> </td>
    </tr>
</xsl:for-each>
</table>
</body></html>
</xsl:template>
</xsl:stylesheet>
```

Web services

A web service is any piece of software that makes itself available over the internet and uses a standardized XML messaging system.

XML is used to encode all communications to a web service. For example, a client invokes a web service by sending an XML message, then waits for a corresponding XML response.

As all communication is in XML, web services are not tied to any one operating system or programming language. Java can talk with Perl. Windows applications can talk with Unix applications.

UDDI

UDDI stands for Universal Description, Discovery and Integration

UDDI is a directory (Yellow pages) for storing information about web services.

UDDI is a directory of web service interfaces described by WSDL

UDDI communicates via SOAP

What is UDDI Based On?

UDDI uses World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF) Internet standards such as XML, HTTP, and DNS protocols.

UDDI uses WSDL to describe interfaces to web services.

Additionally, cross platform programming features are addressed by adopting SOAP, known as XML Protocol messaging specifications-- found at the W3C Web site.

UDDI Benefits

Any industry or businesses of all sizes can benefit from UDDI

Before UDDI, there was no Internet standard for businesses to reach their customers and partners with information about their products and services. Nor was there a method of how to integrate into each other's systems and processes.

Problems the UDDI specification can help to solve:

Making it possible to discover the right business from the millions currently online

Defining how to enable commerce once the preferred business is discovered

Reaching new customers and increasing access to current customers

Expanding offerings and extending market reach

Solving customer-driven need to remove barriers to allow for rapid participation in the global Internet economy

Describing services and business processes programmatically in a single, open, and secure environment

How can UDDI be Used

If the industry published an UDDI standard for flight rate checking and reservation, airlines could register their services into an UDDI directory.

Travel agencies could then search the UDDI directory to find the airline's reservation interface (ie WSDL file).

When the interface is found, the travel agency can communicate with the service using SOAP & WSDL.

WSDL

WSDL stands for Web Services Description Language

WSDL is used to describe web services

WSDL is written in XML

WSDL is a W3C recommendation from 26. June 2007

WSDL Documents

An WSDL document describes a web service.

It specifies the location of the service, and the methods of the service, using these major elements:

Element of WSDL Document:

- 1.types.
2. Message.
3. portType.
4. Binding.
- 5.service.

The main structure of a WSDL document looks like this:

<definitions>

 <types> Data type definitions </types>

 <message> Definition of the data being communicated </message>

 <portType> Set of operations </portType>

<binding>

Protocol (Protocol details of a web services) and data(Message) format specification....

</binding>

<service>

Defines a collection of endpoints in terms of nested port elements that expose bindings.

</service>

</definitions>

types

Defines the (XML Schema) for user defined data types used by the web service.

message

message defines a one-way request or response message

consist of one or more parts (conceptually equivalent to function parameters or return value).

Each part is described by a part element whose name attribute identifies a parameter/return value element.

portType

defines the operations that can be performed, and the messages that are involved.

binding

Protocol (Protocol details of a web services like http) and data(Message) format specification(encoding)....

binding provides details on how a portType operation is transmitted over the wire.

service

Defines a collection of endpoints in terms of nested port elements.

WSDL Example

```
<message name="getTermRequest">
```

```
  <part name="term" type="xs:string"/>
```

```
</message>
```

```
<message name="getTermResponse">
```



```
<part name="value" type="xs:string"/>
</message>
<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
<binding type="glossaryTerms" name="glossaryTermsBinding">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
  <operation>
    <soap:operation soapAction="http://example.com/getTerm"/>
    <input> <soap:body use="literal"/> </input>
    <output> <soap:body use="literal"/> </output>
  </operation>
</binding>
<service name="glossaryTermsService">
  <port name="PQR" binding="">
    <soap:address location="http://localhost:9901/GTS"/>
  </port>
</service>
```

XML SOAP

SOAP stands for Simple Object Access Protocol

SOAP is an application communication protocol

SOAP is a format for sending and receiving messages

SOAP is platform independent

SOAP is based on XML

SOAP is a W3C recommendation

Why SOAP?

It is important for web applications to be able to communicate over the Internet.

The best way to communicate between applications is over HTTP, because HTTP is supported by all Internet browsers and servers. SOAP was created to accomplish this.

SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages.

SOAP Building Blocks

A SOAP message is an ordinary XML document containing the following elements:

- 1) An Envelope element that identifies the XML document as a SOAP message
- 2) Header element that contains header information
- 3) A Body element that contains call or response information
- 4) A Fault element containing errors and status information

Note:

All the elements above are declared in the default namespace for the SOAP envelope: <http://www.w3.org/2003/05/soap-envelope/> and the default namespace for SOAP encoding and data types is: <http://www.w3.org/2003/05/soap-encoding>

Syntax Rules

Here are some important syntax rules:

- 1) A SOAP message **MUST** be encoded using XML
- 2) A SOAP message **MUST** use the SOAP Envelope namespace
- 3) A SOAP message must **NOT** contain a DTD reference
- 4) A SOAP message must **NOT** contain XML Processing Instructions

Skeleton SOAP Message

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

```
<soap:Envelope
```

```
    xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
```

```
    soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
```

```
<soap:Header> Message Header </soap:Header>
```

```
<soap:Body>
```

Body of the Message

<soap:Fault> Error Handling code</soap:Fault>

</soap:Body>

</soap:Envelope>

A SOAP Example

In the example below, a GetStockPrice request is sent to a server. The request has a StockName parameter, and a Price parameter that will be returned in the response. The namespace for the function is defined in "http://www.example.org/stock".

A SOAP request:

POST /InStock HTTP/1.1

Host: www.example.org

Content-Type: application/soap+xml; charset=utf-8

Content-Length: nnn

<?xml version="1.0"?>

<soap:Envelope

xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"

soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">

<m:GetStockPrice>

<m:StockName>IBM</m:StockName>

</m:GetStockPrice>

</soap:Body>

</soap:Envelope>

The SOAP response:

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset=utf-8

Content-Length: nnn

<?xml version="1.0"?>

<soap:Envelope

```

    xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
    soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
<soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPriceResponse>
        <m:Price>34.5</m:Price>
    </m:GetStockPriceResponse>
</soap:Body>
</soap:Envelope>

```

Developing Web Service - JAVA

Defining a Units-Conversion Web Service The units-conversion Web service, consists of four functions for converting between centimeters to inches and degrees Fahrenheit to degrees Celsius. Java API for XML Web Services (JAX-WS) is a standardized API for creating and consuming SOAP web services.

Listing 1: The UC Web service's Service Endpoint Interface(SEI)

```

package ca.javajeff.uc;

import javax.jws.WebMethod;
import javax.jws.WebService;

@WebService
public interface UC
{
    @WebMethod double f2c(double degrees);
    @WebMethod double c2f(double degrees);
}

```

UC describes a Service Endpoint Interface (SEI), which is a Java interface that exposes a Web service interface's operations in terms of abstract Java methods. Clients communicate with SOAP-based Web services via their SEIs. UC is declared to be an SEI via the @WebService annotation.

Listing 2: presents the Web service's UCImpl class(SIB)

```

package ca.javajeff.uc;

import javax.jws.WebService;

```

```

@WebService(endpointInterface = "ca.javajeff.uc.UC")
public class UCImpl implements UC
{
    @Override
    public double c2f(double degrees)
    {
        return degrees * 9.0 / 5.0 + 32;
    }
    @Override
    public double f2c(double degrees)
    {
        return (degrees - 32) * 5.0 / 9.0;
    }
}

```

UCImpl --> describes a Service Implementation Bean (SIB), which provides an implementation of the SEI. This class is declared to be a SIB via the `@WebService(endpointInterface = "ca.javajeff.uc.UC")` annotation. This Web service is ready to be published so that it can be accessed from clients.

Listing 3: presents a UCPublisher

```

import javax.xml.ws.Endpoint;
import ca.javajeff.uc.UCImpl;
public class UCPublisher
{
    public static void main(String[] args)
    {
        Endpoint.publish("http://localhost:9901/UC", new UCImpl());
    }
}

```

Publishing the Web service involves making a single call to the EndPoint class's `publish()` method.

The address parameter identifies the URI assigned to the Web service. Type the URI in the browser you will get the following details about the web service.

Resulting Web page in the Google Chrome Web browser.

Service Name: { http://uc.javajeff.ca/ }UCImplService

Port Name: { http://uc.javajeff.ca/ }UCImplPort

Address: http://localhost:9901/UC

WSDL: http://localhost:9901/UC?WSDL

Implementaion class: ca.javajeff.uc.UCImpl

Listing 4: A client for accessing the UC Web service

```
import java.net.URL;
import javax.xml.namespace.QName;
import javax.xml.ws.Service;
import ca.javajeff.uc.UC;
public class UCClient {
    public static void main(String[] args) throws Exception {
        URL url = new URL("http://localhost:9901/UC?wsdl");
        QName qname = new QName("http://uc.javajeff.ca/", "UCImplService");
        // return Service class instance(ie s1) that provides a client view of a Web service
        Service s1 = Service.create(url, qname);
        qname = new QName("http://uc.javajeff.ca/", "UCImplPort");
        // returns a proxy object ----whose class ---implements the interface UC
        //UC uc = s1.getPort(qname);
        UC uc = s1.getPort(qname, UC.class);
        System.out.printf("DC to DF: 37 DC = %f DF%n", uc.c2f(37.0));
        System.out.printf("DF to DC: 212 DF = %f DC%n", uc.f2c(212.0));
    }
}
```