Programmatic XML

Transentia Pty. Ltd. DonationWare

Tools

- **There are two basic programmer's tools for working with XML**
  - two different styles of processing
    - *Document Object Model (DOM)*
      - originated as a specification to allow JavaScript scripts and Java programs to be portable among Web browsers
      - based around the notion of a document tree
      - W3C standard
        - *level 1 recommended in October 1998*
        - *work on level 2 started immediately afterwards; work on level 3 now underway*
      - actually two sets of interfaces
        - *core*
        - *HTML*
    - *Simple API for XML (SAX)*
      - sees a document as a series of events that may be handled as needed
      - an open-source, community developed system
        - *released May 1998; now in version 2.0*
        - *widely used*

Transentia Pty Ltd. DonationWare

Tree or Event?

- **Which processing style to use?**
  - event-driven processing:
    - *easy*
    - *fast*
    - *restricted*
      - one-pass
        - *no contextual data*
    - *not (yet) a standard*
      - *"A lot of tool vendors will probably implement this API. Although the concrete language to be used (ECMAScript, VBScript, Java, and so on) can differ, you will always deal with the same objects with the same well-known properties and methods. This makes knowledge of the DOM a necessity."*
  - tree processing
    - *fiddly*
    - *resource-hungry*
    - *most powerful*
    - *standardised by the W3C*

Transentia Pty. Ltd. DonationWare

DOM

- **W3C saw a need to:**
  - define
    - *a standardised way to define the logical structure of a document*
    - *how the document is accessed and manipulated*
  - promulgate this standard interface
    - *an end to browser wars?*
      - ho ho ho…
  - XML defines an API
    - *Document Object Model (DOM)*
      - developers can programmatically look at the document as a “tree” with trunks, branches, leaves, etc.
    - *still pretty primitive*
      - doesn’t understand schemas, etc.
      - Java binding != JavaScript binding != CORBA binding, etc.

Transentia Pty. Ltd. DomationWare

DOM Levels

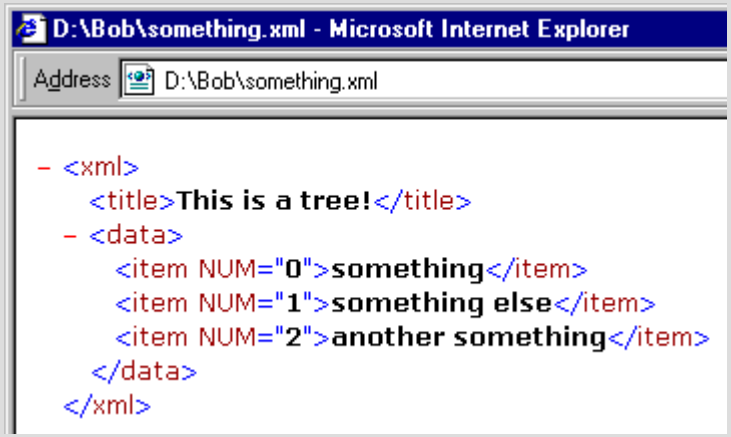
- **From:**
  - http://www.oasis-open.org/cover/dom.html

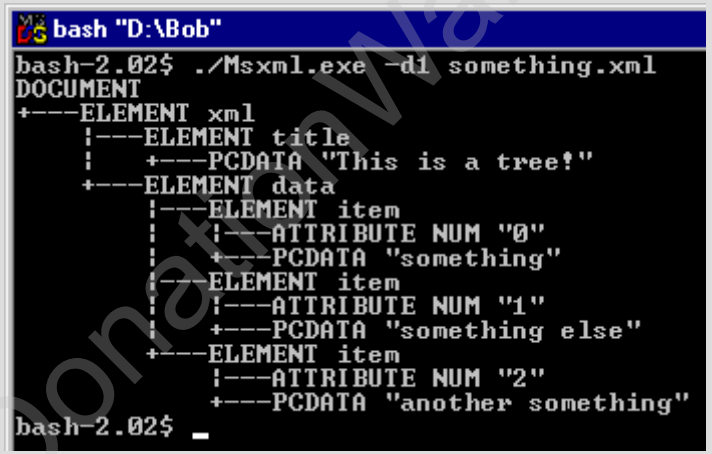
'DOM is being designed at several levels:

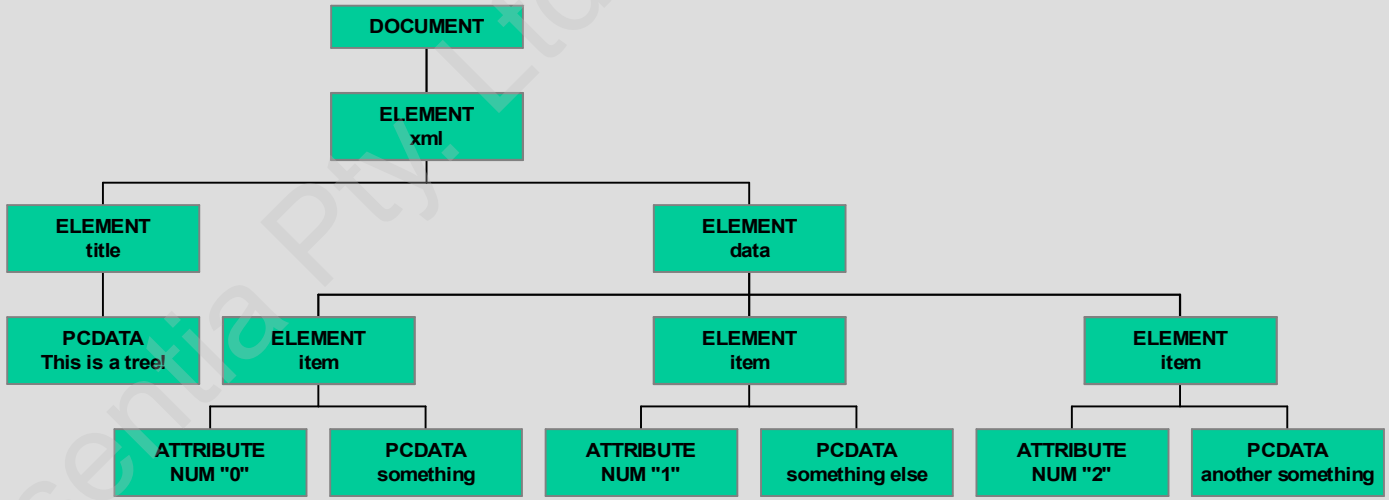
- "Level 1. This concentrates on the actual core, HTML, and XML document models. It contains functionality for document navigation and manipulation.
- Level 2 *Recommendation as of 13 Nov. 2000.* Includes a style sheet object model, and defines functionality for manipulating the style information attached to a document. It also enables traversals on the document, defines an event model and provides support for XML namespaces.
- Level 3 *First public working drafts are available.* Will address document loading and saving, as well as content models (such as DTDs and schemas) with document validation support. In addition, it will also address document views and formatting, key events and event groups.
- Further Levels. These may specify some interface with the possibly underlying window system, including some ways to prompt the user. They may also contain a query language interface, and address multithreading and synchronization, security, and repository."'

Trial Version Pty. Ltd. DonationWare

Document == Tree







Transentia Pty Ltd. DonationWare

Simple API

- **DOM exposes a very simple API**
  - four *base objects* defined:
    - *Document*
    - *Node*
      - base object for all entries in a document
    - *NodeList*
      - represent an ordered collection of Nodes
    - *NamedNodeMap*
      - an unordered collection of Nodes
        - *similar to a dictionary*
  - IE5 extends these objects
    - *as do all other actual implementations*
      - each in their own way
        - *so much for 'standard'…*
    - *makes them more useful*
      - makes them fit into a 'platform' better

Transentia Pty Ltd. DomationWare

Node API

- **A single node in the document tree**
  - methods include:
    - *hasChildNodes*
    - *insertBefore, removeChild, replaceChild, appendChild*
    - *cloneNode*
    - *createAttribute, createCDATASection, createComment, createDocumentFragment, createElement, createEntityReference, createProcessingInstruction, createTextNode*
    - *getElementsByTagName*
  - some properties:
    - *nodeName, nodeType, nodeValue*
    - *attributes, documentElement, ownerDocument, doctype*
    - *childNodes, firstChild, lastChild, nextSibling, parentNode, previousSibling*

Transcentia Ltd. DonationWare

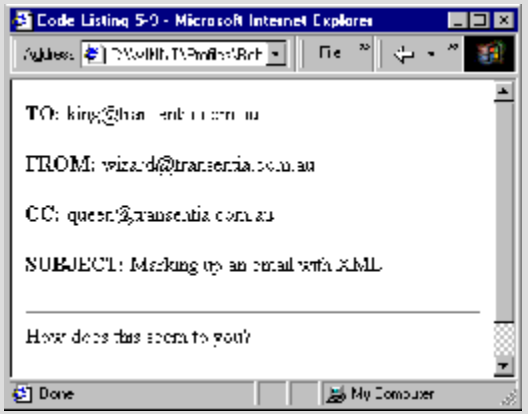
Another IE5 Example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript" FOR=window EVENT=onload>
      loadDoc()
    </SCRIPT>
    <SCRIPT LANGUAGE="JavaScript">
      var xmlDoc = new ActiveXObject("microsoft.xmldom");
      xmlDoc.load("Lst5_3.xml");

      function loadDoc()
        {
        if (xmlDoc.readyState == "4")
          start()
        else
          window.setTimeout("loadDoc()", 4000);
        }

      function start()
        {
        var newHTML = "";
        rootElem = xmlDoc.documentElement;
        for (el=0; el<rootElem.childNodes.length; el++)
          if (el != rootElem.childNodes.length-1)
            newHTML = newHTML +
              "<SPAN STYLE='font-weight:bold;font-size:16'>" +
              rootElem.childNodes.item(el).nodeName +
              ": </SPAN><SPAN STYLE='font-weight:normal'>" +
              rootElem.childNodes.item(el).text + "</SPAN><P>";
          else
            newHTML = newHTML +
              "<HR><SPAN STYLE='font-weight:normal'>" +
              rootElem.childNodes.item(el).text + "</SPAN><P>";
        content.innerHTML = newHTML;
        }
    </SCRIPT>
    <TITLE>Code Listing 5-9</TITLE>
  </HEAD>
  <BODY>
    <DIV ID="content"></DIV>
  </BODY>
</HTML>
```

```
<?xml version="1.0"?>
<EMAIL>
  <TO>king@transentia.com.au</TO>
  <FROM>wizard@transentia.com.au</FROM>
  <CC>queen@transentia.com.au</CC>
  <SUBJECT>Marking up an email with XML</SUBJECT>
  <BODY>How does this seem to you?</BODY>
</EMAIL>
```



Transentia Pty Ltd. DocumentWare

Node Extensions

- **All extend Node:**
  - Document
  - Attribute
  - CDATASection
  - CharacterData
  - Comment
  - DocumentFragment
  - DocumentType
  - Element
  - Entity
  - EntityReference
  - Implementation
  - Notation
  - ProcessingInstruction
  - Text
  - the NodeType enumeration provides values to allow each node type to be distinguished
  - each also has a nodeString property which gives its name
    - *"attribute", "entity" "processingInstruction", etc.*

```
function getNodeType(intType) {
  switch (intType) {
    case 1: return "ELEMENT (1)"; break;
    case 2: return "ATTRIBUTE (2)"; break;
    case 3: return "TEXT (3)"; break;
    case 4: return "CDATA SECTION (4)"; break;
    case 5: return "ENTITY REFERENCE (5)"; break;
    case 6: return "ENTITY (6)"; break;
    case 7: return "PROCESSING INSTRUCTIONS (7)"; break;
    case 8: return "COMMENT (8)"; break;
    case 9: return "DOCUMENT (9)"; break;
    case 10: return "DOCUMENT TYPE (10)"; break;
    case 11: return "DOCUMENT FRAGMENT (11)"; break;
    case 12: return "NOTATION (12)";
   }
}
```

Transentia Pty. Ltd. Donation Ware

NodeList API

- **A 'live' collection of Nodes**
  - addition and removal of nodes, and changes within nodes, are immediately reflected in the collection
  - methods
    - *item*
  - properties
    - *length*

Transentia Pty. Ltd. DonationWare

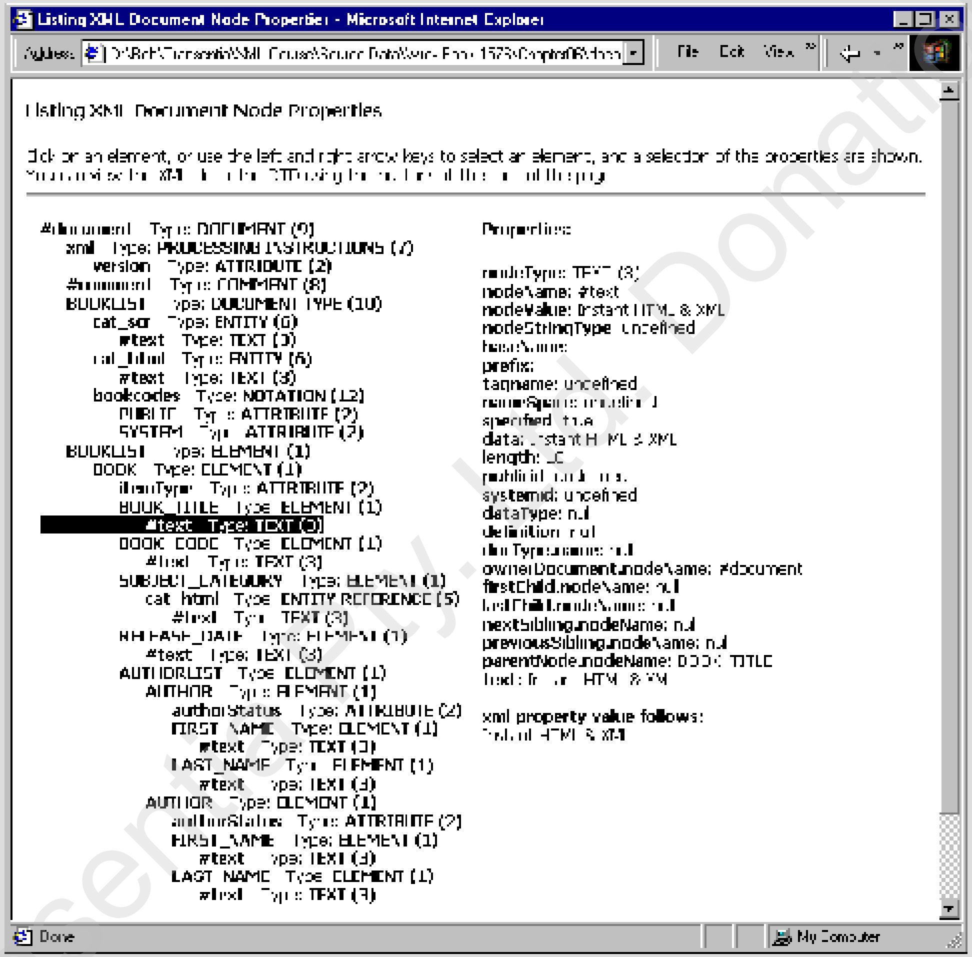
NamedNodeMap API

- **Provides access to attributes by attribute name**
  - also live
  - unordered
    - *although also allows indexed access*
  - methods
    - *item, getNamedItem, removeNamedItem. setNamedItem*
  - properties
    - *length*

Transentia Pty. Ltd. DonationWare

Node Properties

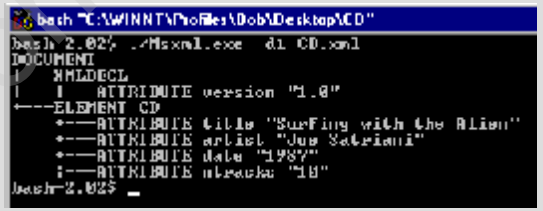
- **An example**
  - from the Wrox "XML IE5" book...



Transcend... DonationWare

NamedNodeMap Processing

```
<?xml version="1.0"?>
<CD title="Surfing with the Alien"
    artist="Joe Satriani"
    date="1987"
    ntracks="10" />
```

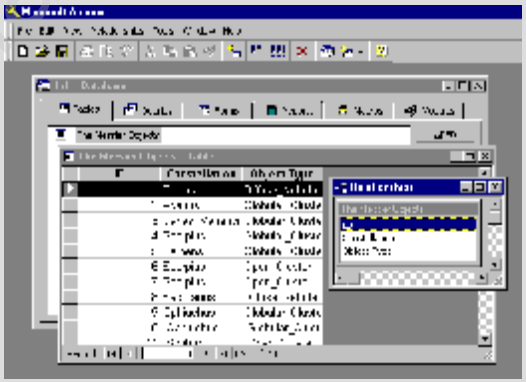



```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
  <head>
    <title>NamedNodeMap Processing</title>
    <xml id="CD" src="CD.xml"></xml>
    <script for=window event=onload>
      <!--
      div.innerHTML = "";
      var xmlDoc = document.all("CD").XMLDocument;
      var root = xmlDoc.documentElement
      var attrs = root.attributes;
      div.innerHTML += "<strong>Title: </strong>" +
                       attrs.getNamedItem("title").text + "<br>";
      div.innerHTML += "<strong>Artist: </strong>" +
                       attrs.getNamedItem("artist").text + "<br>";
      div.innerHTML += "<strong>Date: </strong>" +
                       attrs.getNamedItem("date").text + "<br>";
      div.innerHTML += "<strong>Number of tracks: </strong>" +
                       attrs.getNamedItem("ntracks").text + "<br>";
      -->
    </script>
  </head>
  <body>
    <p>
      <DIV ID="div"></DIV>
    </p>
  </body>
</html>
```

Transentia Pty. Ltd. Donation Ware

DOM Example

- **Creating XML Using DOM**
  - from a very simple SQL database
  - involves:
    - *using Java/JDBC to access the database*
      - the examples use the (sadly, now defunct) DataChannel Java Parser 'XJParser'
        - *could easily use C++/Python/PERL, etc.*
    - *building a DOM tree structure in memory*
      - establish the relevant Document Type to ensure that the produced XML can be validated
    - *outputting this tree*
  - the database is a very simple Microsoft Access table



Transentia Pty. Ltd. DonationWare

- **A simple Java program:**

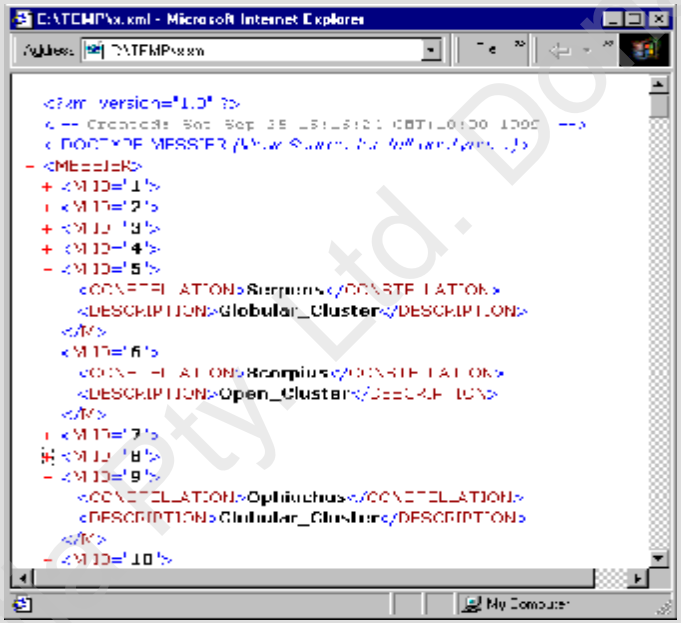
```
import java.sql.*;
import com.datachannel.xml.om.*;

public class SQLDumper
  {
  static
    { try { Class.forName ("com.ms.jdbc.odbc.JdbcOdbcDriver"); } catch (Exception e) {} }
  public static void main (String [] args) throws Exception
    {
    Document doc = new Document ();
    doc.appendChild (doc.createProcessingInstruction ("xml", "version=\"1.0\""));
    doc.appendChild (doc.createComment ("Created: " + new java.util.Date ()));
    IXMLDOMElement root = (IXMLDOMElement) doc.createElement ("MESSIER");
    DocumentType.createDocumentType (doc, "MESSIER SYSTEM \"MessierDTD.dtd\"");
    doc.appendChild (root);
    Statement statement =
      DriverManager.getConnection ("jdbc:odbc:The Messier Database",
                                   "Messier", "MessierMan").createStatement ();
    ResultSet rs = statement.executeQuery ("SELECT * FROM [The Messier Objects]");
    while (rs.next ())
      {
      int id = rs.getInt ("ID");
      IXMLDOMElement child = (IXMLDOMElement) doc.createElement("M");
      child.setAttribute ("ID", "" + id);
      IXMLDOMElement c = (IXMLDOMElement) doc.createElement ("CONSTELLATION");
      c.appendChild (doc.createTextNode (rs.getString ("Constellation")));
      child.appendChild (c);
      IXMLDOMElement d = (IXMLDOMElement) doc.createElement ("DESCRIPTION");
      d.appendChild (doc.createTextNode(rs.getString ("Object Type")));
      child.appendChild (d);
      root.appendChild (child);
      }
    System.out.println (doc.getXML());
    }
  }
```

Transsoft Corporation Ltd. - DonationWare

Creating XML...

- **The output can be shown in IE5:**
  - well-formed
  - valid
    - *linked to an external DTD*

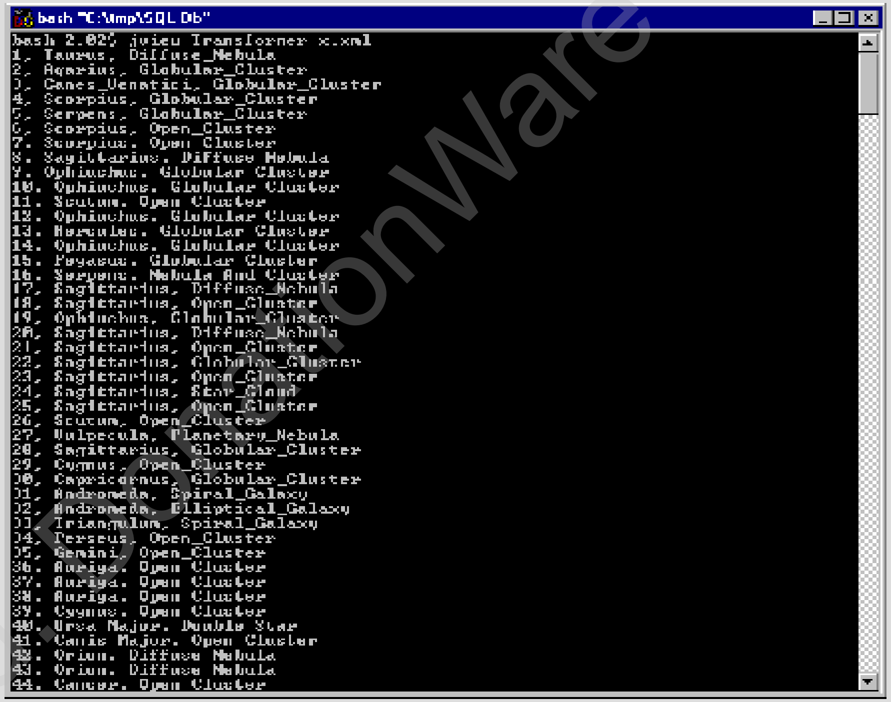


Transentia Pty. Ltd. DonationWare

- **A simple Java program to produce a comma delimited representation of XML-formatted data**
  - involves:
    - *loading XML data into DOM*
      - the XML file just produced, in this example…
    - *walking the DOM tree, producing the comma delimited file as output*

Transentia Pty. Ltd. DonationWare

Transforming XML...



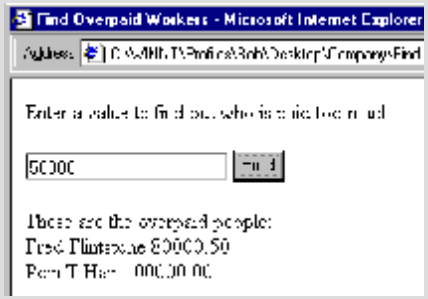

```
import com.datachannel.xml.om.*;

public class Transformer
  {
  public static void main (String [] args)
    {
    Document doc = new Document ();
    doc.load (args [0]);
    IXMLDOMElement root =
      (IXMLDOMElement) doc.getDocumentElement ();
    IXMLDOMNodeList nodelist =
      (IXMLDOMNodeList) root.getChildNodes ();
    for (int i = 0; i < nodelist.getLength (); i ++)
      {
      IXMLDOMNode n = (IXMLDOMNode) nodelist.item (i);
      IXMLDOMNodeList nl = (IXMLDOMNodeList) n.getChildNodes ();
      IXMLDOMNamedNodeMap attributes = (IXMLDOMNamedNodeMap) n.getAttributes ();
      IXMLDOMNode IDNode = (IXMLDOMNode) attributes.getNamedItem ("ID"),
                  constellation = (IXMLDOMNode) n.getFirstChild (),
                  description = (IXMLDOMNode) n.getLastChild ();
      System.out.println (IDNode.getText () + ", " + constellation.getText () +
                          ", " + description.getText ());
      }
    }
  }
```

IE5 Examples

- IE5 supplies scripting interfaces to the DOM

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
  <head>
    <title>Find Overpaid Workers</title>
    <xml id="Company" src="Company.xml"></xml>
    <script>
      <!--
      function find()
        {
        overPaid.innerHTML = "";
        var xmlDoc = document.all ("Company").XMLDocument;
        var root = xmlDoc.documentElement;
        var kids = root.getElementsByTagName ("emp");
        var value = inputText.value * 1.0;
        for (var i = 0; i < kids.length; i ++)
          {
          var node = kids.item(i);
          var salaries = node.getElementsByTagName ("salary");
          var salaryNode = salaries.item(0);
          var salaryValue = salaryNode.childNodes.item(0).nodeValue * 1.0;
          if (salaryValue >= value)
            overPaid.innerHTML += node.text + "<br>";
          }
        }
      -->
    </script>
  </head>
  <body>
    <p>
      Enter a value to find out who is paid too much:
    </p>
    <p>
      <input type="text" name="inputText" size="20">
      <input type="button" name="findButton" value="Find" onClick="find()">
    </p>
    <p>
      These are the overpaid people: <DIV ID="overPaid"></DIV>
    </p>
  </body>
</html>
```



Transcentia Pty. Ltd. DonationWare

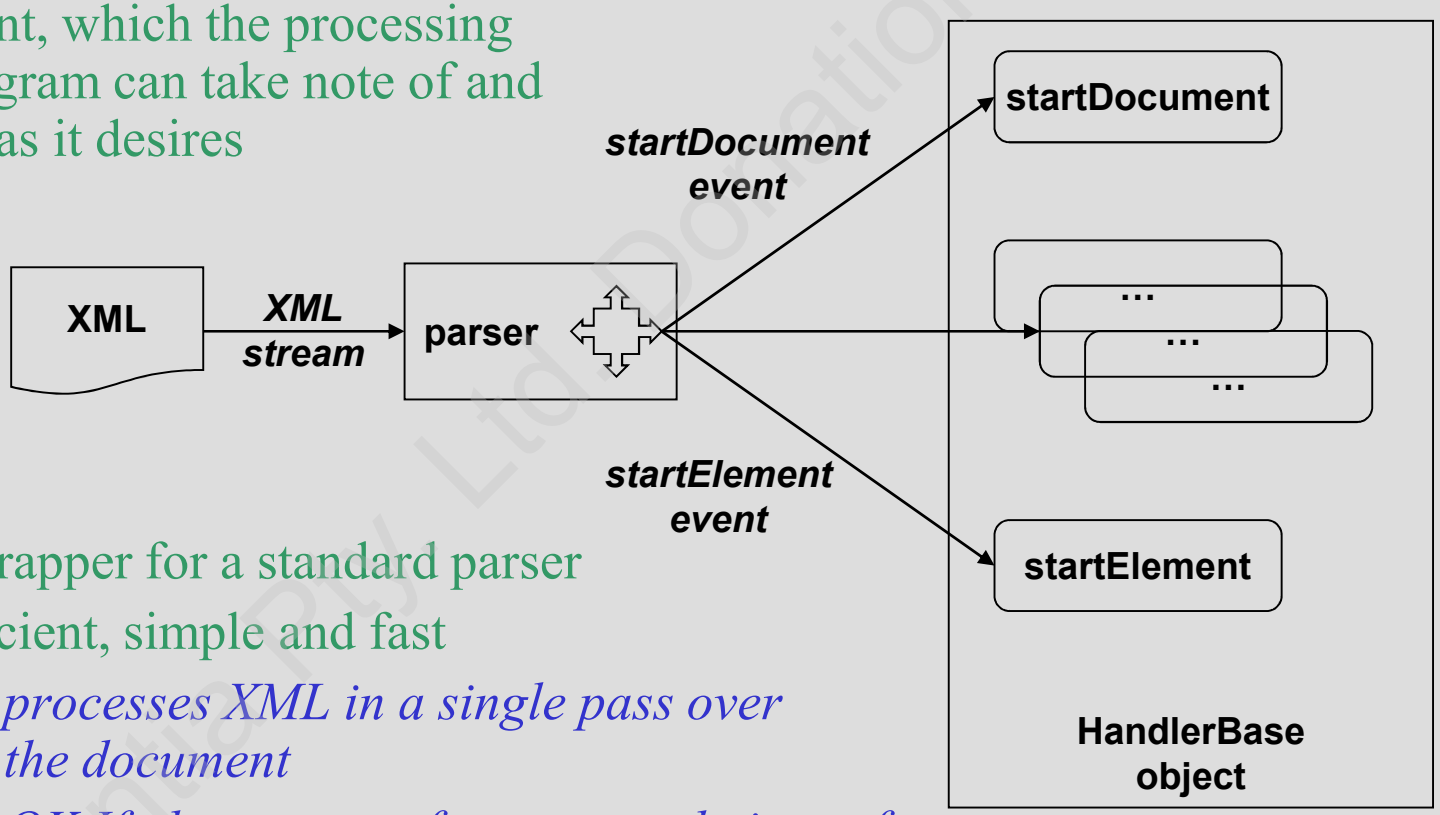
```
function showBirthdate()
  {
  var root = employeeDoc.documentElement;
  var selectedElems = root.selectNodes("employee[name='Mary Davis']");
  var maryElem = selectedElems.item(0);
  var birthdate = maryElem.childNodes.item(1).nodeTypedValue;
  alert("Mary Davis was born " + birthdate);
  }
```

```
<HTML>
<HEAD><TITLE>Elements: Hierarchy</TITLE>
<SCRIPT LANGUAGE="JScript">
function showHierarchy() {
    var depth = 0;
    var msg = document.all(0).tagName;
    for (i=1; i<document.all.length; i++) {
        if (document.all(i-1).contains(document.all(i))==true) {
            depth = depth + 1;
        } else {
            var elParent = document.all(i-1).parentElement;
            for ( ; depth>0; depth--) {
                if (elParent.contains(document.all(i))==true)
                    break;
                elParent = elParent.parentElement;
            }
        }
        msg = msg + "\n";
        for (j=1; j<=depth; j++)
            msg = msg + "  ";
        msg = msg + document.all(i).tagName;
    }
    alert("This document contains:\n" + msg);
}
</SCRIPT>
</HEAD>
<BODY onload="showHierarchy()">
<H1>Welcome!</H1>
<P>This document is <B>very</B> short.
</BODY>
</HTML>
```

More...
Transcentia Pty. Ltd. DonationWare

SAX

- **SAX (Simple API for XML) is an event-oriented system**
  - stream-oriented; each tag encountered generates an event, which the processing program can take note of and act as it desires



  - a wrapper for a standard parser
  - efficient, simple and fast
    - *processes XML in a single pass over the document*
    - *OK If the context of a processed piece of data within the overall XML document is unimportant*
  - typically used where the underlying document does not need to be modified

TransentiaPty Ltd DonationWare

- **The API is relatively simple**
  - five interfaces relevant to application writers
    - *DocumentHandler*
      - main interface: application creates a class that implements this interface and registers an instance with the SAX parser
        - *or, may extend HandlerBase convenience class*
      - methods:
        - *startDocument, endDocument*
        - *startElement, endElement*
        - *characters*
        - *processingInstruction*
    - *AttributeList*
      - allows processing of an element's attributes
    - *ErrorHandler*
      - for customized error handling
    - *DTDHandler*
      - permits the processing of notations and unparsed entities
    - *EntityResolver*
      - lets application writer provide customized handling for external entities

Transsentia Pty. Ltd. Domain Ware

SAX 1.0 Example...

- **A Simple SAX application:**
  - scans the XML document to establish how many galaxies were included in Charles Messier's list
    - *establishes a Handler for the events that the SAX parser generates*
      - arranges to be informed whenever start and end tags are seen and when the whole document has been parsed
      - looks out for DESCRIPTION elements, and examines the characters that comprise the body of the element

Transentia Pty. Ltd. DonationWare

SAX 1.0 Example (Cont'd)

```
<?xml version="1.0"?>
<MESSIER xmlns="x-schema:MessierSchema.xml">
  <M INDEX="1">
    <CONSTELLATION>Taurus</CONSTELLATION>
    <DESCRIPTION>Diffuse Nebula</DESCRIPTION>
  </M>
  <M INDEX="2">
    <CONSTELLATION>Aqarius</CONSTELLATION>
    <DESCRIPTION>Globular Cluster</DESCRIPTION>
  </M>
  <M INDEX="3">
    <CONSTELLATION>Canes Venatici</CONSTELLATION>
    <DESCRIPTION>Globular Cluster</DESCRIPTION>
  </M>
  ...
</MESSIER>
```

```
import org.xml.sax.*;
import org.xml.sax.helpers.ParserFactory;

class MessierHandler extends HandlerBase
  {
  private static final String descTagName = "DESCRIPTION";
  private int nGalaxies = 0;
  private boolean inDescriptionElement = false;
  public void endDocument ()
    {
    System.out.println ("There are " + nGalaxies + " Messier galaxies.");
    }
  public void startElement (String name, AttributeList atts)
    {
    inDescriptionElement = name.equals (descTagName);
    }
  public void endElement(String name)
    {
    inDescriptionElement = ! name.equals (descTagName);
    }
  public void characters (char ch [], int start, int length)
    {
    if (inDescriptionElement && new String (ch, start, length).endsWith ("alaxy"))
      nGalaxies ++;
    }
  }

public class SAXCount
  {
  public static void main (String [] args) throws Exception
    {
    Parser parser = ParserFactory.makeParser ("com.ibm.xml.parsers.SAXParser");
    parser.setDocumentHandler (new MessierHandler ());
    parser.parse (args [0]);
    }
  }
```

```
bash "C:\TEMP\SQL Db"
bash-2.02$ jview SAXCount x.xml
There are 39 Messier galaxies.
bash-2.02$
```

Transentia Pty. Ltd. Donation Ware

SAX 2.0

- SAX is now at version 2.0
- Major changes include
  - bugfixes/better documentation
  - replacing HandlerBase class with DefaultHandler
  - namespace support
  - support for chains of filters
  - configurable XMLReaders
    - *provides methods for setting/getting parser properties*

```
try {
  if (xmlReader.getFeature("http://xml.org/sax/features/validation"))
{
    System.out.println("Parser is validating.");
  } else {
    System.out.println("Parser is not validating.");
  }
} catch (SAXException e) {
  System.out.println("Parser may or may not be validating.");
}
```

  - extension packages
  - more widespread support from parser vendors
    - *Sun's JAXP 1.1 supports SAX 2.0*

Transoniq Pty. Ltd. DonationWare

JAXP

- **Sun's "Java API for XML Processing"**
  - developed under auspices of Java Community Process
  - enables applications to parse and transform XML documents independent of a particular implementation
  - gives the flexibility to swap between XML processors (such as high performance vs. memory conservative parsers) without making application code changes
    - *interfaces with DOM and SAX*

```
public static void main(String[] args) {
  try {
    // Get SAX Parser Factory
    SAXParserFactory factory = SAXParserFactory.newInstance();

    // Turn on validation, and turn off namespaces
    factory.setValidating(true);
    factory.setNamespaceAware(false);

    SAXParser parser = factory.newSAXParser();
    parser.parse(new File(args[0]), new MySAXHandler());

    } catch (ParserConfigurationException e) {
      System.out.println("The underlying parser does not support the requested features.");
    } catch (FactoryConfigurationError e) {
      System.out.println("Error occurred obtaining SAX Parser Factory.");
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
```

```
prompt% java -Djavax.xml.parsers.SAXParserFactory-…
```

Transentia Pty. Ltd. Domain Vault

Java XML Pack

- **Sun's latest foray in the XML world**
- **Includes:**
  - Java API for XML Messaging (JAXM)
  - Soap with Attachments API for Java (SAAJ)
  - Java API for XML Processing (JAXP)
  - Java API for XML Registries (JAXR)
  - Java API for XML-based RPC (JAX-RPC)
- **A foundation for other things**
  - e.g. Java Web Services Developer Pack

Transentia Pty. Ltd. DonationWare

JAXB

- **Java Architecture for XML Binding**
  - currently in Early Access from Sun
  - an API and tools that automate the mapping between XML documents and Java objects
    - *generated classes may be more efficient than SAX/DOM*
      - more specific than a generic parser
  - intended to permit generic (un)marshalling to almost anything
    - *XML, database, etc.*
- **Other mature implementations of this sort of technology exist**
  - ExoLabs *Castor*
    - Castor XML: Java object model to and from XML
      - *Generate source code from an XML Schema*
    - Castor JDO: Java object persistence to RDBMS
    - Castor DAX: Java object persistence to LDAP
    - Castor DSML: LDAP directory exchange through XML
    - etc.

"To make your Java programming life even easier, keep an eye on both JAXB and Castor for the next wave of exciting advances."

Transcentia Pty. Ltd. Donation Ware

JAXB Example

```
import java.io.*;
import java.util.*;
import java.math.BigDecimal;
public class JAXBSocksTest
{
  public static void main(String[] args) {
    Socks socks = new Socks();
    try {
      File socksFile = new File(args[0]);
      InputStream fin = new FileInputStream(socksFile);
      socks = socks.unmarshal(fin);
      List sockList = socks.getSock();
      fin.close();
      Sock sock;
      Color black; // Color class generated by JAXB
      for (Iterator i = sockList.iterator(); i.hasNext();) {
        sock = (Sock)i.next();
        if (sock.getColor().
              getValue().equals(SockColor.WHITE)) {
          black = new Color();
          black.setValue(SockColor.BLACK);
          sock.setColor(black);
        }
      }
      sock = new Sock();
      sock.setNumber("4");
      sock.setName("new sock");
      sock.setImage("newsock.jpg");
      Color c = new Color();
      c.setValue(SockColor.BLACK);
      sock.setColor(c);
      sock.setSmell("0");
      sock.setPrice(new BigDecimal("5.55"));
      sockList.add(sock);
      socks.validate();
      File socks_new = new File("blackSocks.xml");
      FileOutputStream
        fout = new FileOutputStream(socks_new);
      socks.marshal(fout);
      fout.close();
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
```

Javaworld example

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="socks">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="sock" minOccurs="0"
                     maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="name" type="xsd:string"/>
              <xsd:element name="image" type="imageType"/>
              <xsd:element name="color" type="colorType"/>
              <xsd:element name="price" type="money"/>
              <xsd:element name="smell" type="smellType"/>
              </xsd:sequence>
              <xsd:attribute name="number"
                             type="xsd:string" use="required"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:simpleType name="imageType">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="(.)+\.(gif|jpg|jpeg|bmp)"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="colorType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="black"/>
      <xsd:enumeration value="white"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="money">
    <xsd:restriction base="xsd:decimal">
      <xsd:fractionDigits value="2"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="smellType">
    <xsd:annotation>
      <xsd:documentation>0=clean, 10=terrible</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:nonNegativeInteger">
      <xsd:minInclusive value="0"/>
       <xsd:maxInclusive value="10"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

Transentia Pty. Ltd. Domain Ware