# Data Importer

# The Main Routine

- In these notes I call it DataImporter.  It contains:
  - A main routine the creates an instance of IndexerData, the root of our data model.
  - It has two supplemental static methods that make our job much easier.
    - ArrayList<Element> getChildElements(Node)
    - String getValue(Element)

# Setting It Up

```
// When in conflict use the class from org.w3c.dom.
//       Examples: Document, Element, Node, NodeList

File xmlFile = new File("Records.xml");

DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
Document doc = dBuilder.parse(xmlFile); //Can use URI instead of xmlFile

//optional, but recommended. Read this
//       http://stackoverflow.com/questions/13786607/normalization-in-dom-parsing-with-java-how-does-it-work
doc.getDocumentElement().normalize();

Element root = doc.getDocumentElement();

indexerData = new IndexerData(root);
```

# getChildElements (not needed if you only use getElementsbyTagName(String))

```
public static ArrayList<Element> getChildElements(Node node) {
    ArrayList<Element> result = new ArrayList<Element>();

    NodeList children = node.getChildNodes();
    for(int i = 0; i < children.getLength(); i++) {
       Node child = children.item(i);
       if(child.getNodeType() == Node.ELEMENT_NODE){
          result.add((Element)child);
       }
    }

    return result;
}
```

# getValue

```
public static String getValue(Element element) {
    String result = "";
    Node child = element.getFirstChild();
    result = child.getNodeValue();
    return result;
}
```

# The Class IndexerData

```
//An Example of how to use “DataImporter.getChildElements(Element)
public class IndexerData {
    private ArrayList<User>     users      = new ArrayList<User>();
    private ArrayList<Project> projects = new ArrayList<Project>();

    public IndexerData(Element root) {
       ArrayList<Element> rootElements = DataImporter.getChildElements(root);

       ArrayList<Element> userElements =
          DataImporter.getChildElements(rootElements.get(0));
       for(Element userElement : userElements) {
          users.add(new User(userElement));
       }

       ArrayList<Element> projectElements =
          DataImporter.getChildElements(rootElements.get(1));
       for(Element projectElement : projectElements) {
          projects.add(new Project(projectElement));
       }
    }
}
```

# Using “getElementsByTagName” and “DataImporter.getValue(Element),

```
public class Project {
      private String title = null;
      private int recordsPerImage = -1;
      private int firstYCoord = -1;
      private int recordHeight = -1;

      ArrayList<Field> fields = new ArrayList<Field>();
      ArrayList<Image> images = new ArrayList<Image>();

    public Project(Element projectElement) {
        title = DataImporter.getValue((Element)projectElement.getElementsByTagName("title").item(0));
        recordsPerImage = Integer.parseInt(DataImporter.getValue(
                            (Element)projectElement.getElementsByTagName("recordsperimage").item(0)));
        firstYCoord = Integer.parseInt(DataImporter.getValue(
                            (Element)projectElement.getElementsByTagName("firstycoord").item(0)));
        recordHeight = Integer.parseInt(DataImporter.getValue(
                            (Element)projectElement.getElementsByTagName("recordheight").item(0)));

        Element fieldsElement = (Element)projectElement.getElementsByTagName("fields").item(0);
        NodeList fieldElements = fieldsElement.getElementsByTagName("field");
        for(int i = 0; i < fieldElements.getLength(); i++) {
            fields.add(new Field((Element)fieldElements.item(i)));
        }

        Element imagesElement = (Element)projectElement.getElementsByTagName("images").item(0);
        NodeList imageElements = imagesElement.getElementsByTagName("image");
        for(int i = 0; i < imageElements.getLength(); i++) {
            images.add(new Image((Element)imageElements.item(i)));
        }
    }
}
```