

Assignment #7: XML – Phone Book Services

DUE: 9:30 am, Monday, May 31

Overview

In this assignment, you will work with a collection of assigned phone numbers stored in an XML document. You will use various XML technologies to perform certain tasks: (i) a DTD for defining valid phone book xml documents (ii) an XSLT description for creating an HTML view of the phone book (iii) a SAX application to answer simple queries on the phone book data.

Part A: DTD

Write a DTD to define the structure of an XML document representing the information stored by the phone company for a client. Your DTD should support the following aspects:

1. A client can have a single name, a first and last name, or a first name, last name, and (potentially multiple) middle names.
2. A client has a single address, but can have several numbers
3. A client need not have a recorded address
4. Each client must have a unique customer identifier, and is designated either as a business or as an individual.
5. A client may optionally be designated as “unlisted” (meaning they will not appear in any listing); by default clients are included in listings.
6. An address (if given) may optionally be designated as “private”.

You may find it helpful to develop the DTD along with an example XML document of the desired type. The most convenient way to do this is by embedding the DTD definition directly in the header of the document. That is, you can write something like:

```
<?xml version="1.0"?>
<!DOCUMENT phone_book [
  <!ELEMENT ... etc ... >
  <!ATTLIST ... etc ... >
]>

<!-- your xml document goes here -->
```

This way you can gradually refine your DTD, as you extend and expand your example. Validate your document as you go, possibly by using the online XML document validator indicated on the class web page (under XML in the resource page).

Be sure your final submission of your DTD is as a separate document, however, not embedded in some example XML document.

Part B: XSLT

Use XSLT to create an HTML page from a phone book XML document (conformant to your DTD from Part A). The resulting HTML document should have (at least) the following features:

1. A heading “Phone Book Listing”.
2. A table, with borders, containing the listing data (names, addresses, and numbers only).
3. Each row of the table should correspond to a single customer.
4. Unlisted customers should not appear
5. A private address should not appear

You should use the Xalan XSLT processor, installed at `/usr/class/cis894/Xalan`. The documentation for this processor can be found on the resource page at the class web site. There are some useful examples under the installation directory.

The HTML should be generated by running `xalan.xslt.Process` directly (ie you should not use the `xml-stylesheet` processing instruction as this is not well-supported by Netscape). For example, we expect to run:

```
% setenv XALAN_ROOT /usr/class/cis894/Xalan
% setenv XALAN_PATH ${XALAN_ROOT}/bin/xalan.jar:${XALAN_ROOT}/bin/xml-apis.jar:
${XALAN_ROOT}/bin/xercesImpl.jar
% java -classpath $XALAN_PATH org.apache.xalan.xslt.Process -in book1.xml
-xsl phone_book.xsl -out book1.html
```

Part C: SAX

Write a Java application for phone number look-up. Your application should run from the command line, taking a single argument: the name of the XML document containing the customer data. Your application should support both lookup and reverse lookup functionality. For the former, the user should be prompted to enter a target last name and the application should retrieve the corresponding number(s). For the latter, the user should be prompted to enter a target number and the application should retrieve the corresponding last name. For simplicity, your application can ignore the “unlisted” aspect of customer data (ie treat all customer entries as retrievable).

Your application should use the SAX API, as provided by the Xerces XML parser, installed at `/usr/class/cis894/Xerces`. The documentation for this parser can be found on the resource page at the class web site. There are some useful examples under the installation directory.

You will probably compile your program with something like the following:

```
% setenv XERCES_ROOT /usr/class/cis894/Xerces
% javac -classpath .:${XERCES_ROOT}/classes Finder.java
```

And run the program with:

```
% java -classpath .:${XERCES_ROOT}/classes Finder book1.xml
```

What to turn in

You should submit the following:

1. An XML document, named `book1.xml`, representing the phone customer databases. Your example document should be large and varied enough to adequately illustrate all the aspects of its DTD.
2. The DTD from part A, named `phone_book.dtd`.
3. The XSLT document from part B, named `phone_listing.xsl`.
4. Your Java program from part C.
5. A readme file, named `README`, detailing how to compile and run your Java SAX program.

Use the usual submit command to turn these things in. No hard-copy submission is required.