CMPSC 380
Principles of Database Systems
Fall 2014

Laboratory Assignment Eight
Using the Document Object Model to Parse XML Files

Introduction

Sections 23.4 and 23.5 of your textbook explain the steps that a software developer and/or a data analyst must take to query, transform, and parse eXtensible markup language (XML) documents. In this laboratory assignment, you will download and use two programs that leverage the tree-based document object model (DOM) to parse an XML document. In addition, you will extend the second program so that it can produce output in a more configurable fashion. Finally, you and your partner will conduct a simple experiment to measure the performance of DOM-based parsing and write a short report that describes and explains the empirical trends that you identified.

Learning More About XML Parsing Methods

In addition to reviewing the aforementioned sections of your textbook—and any other relevant content in Chapter 23—you should study the following Web site: `http://www.mkyong.com/java/how-to-read-xml-file-in-java-dom-parser/`. We will use the simple example provided in this tutorial to start learning about how to parse an XML document. You should also carefully review Oracle's tutorial, called "Reading XML Data into a DOM", which is available from `https://docs.oracle.com/javase/tutorial/jaxp/dom/readingXML.html`. To gain a better understanding of the different DOM-based solutions that are available and to ensure that you understand some fundamental trade-offs associated with using our chosen DOM parser, you should also read `https://docs.oracle.com/javase/tutorial/jaxp/dom/when.html`.

Trying Simple DOM-Based Parsing of XML

You and your partner should create a Git repository, hosted by Bitbucket, that you can use to complete this laboratory assignment. To start off this project, one of you should go into the "share" repository for this course and run the "`git pull`" command. After investigating the directories for this course, you will find the "`lab8/`" directory. What files does this directory contain? Please see the course instructor if your repository does not have two XML files and two Java programs.

You are your partner should use a text editor, like GVim, to view the file called "`staff.xml`". What are the contents of this file? Next, you should use GVim to customize the contents of this file so that they describe staff members at Allegheny College—you can take a guess for any attributes of a staff member that you do not know. In addition, you are encouraged to add in additional data and meta-data as long as it adheres to the standard set forth in the original version of "`staff.xml`". Now, you should compile and run the program called "`ReadXMLFileWithDOM.java`". How does this program work? What is the output of this program? Does the program seem to work correctly? What are the limitations associated with this program's implementation?

Using and Extending a Comprehensive XML Parser

After completing the previous phase of the laboratory assignment, you and your partner should again use GVim to study the contents of the provided "`nih-mesh.xml`" file, available from `http://www.nlm.nih.gov/mesh/filelist.html`. As you will notice, this XML document is substantially more complex than the "`staff.xml`" file that you previously parsed. When you try to parse this new XML file using the "`ReadXMLFileWithDOM.java`" program, does it work correctly?

The "`DOMEcho.java`" program provides another, more comprehensive, approach to parsing an XML document. Please run this new parser on both of the XML files in the Git repository. What output does this program produce? Does it seem to work correctly? One of the downsides of the "`DOMEcho`" program is that it always produces the debugging output that shows the details about each XML component—a feature that is not desirable if we want to benchmark the performance of DOM-based parsing. You and your partner should add a command-line argument that will turn off all of this program's debugging output, while ensuring that it still parses the entire XML document. Next, you should add code that appropriately times the execution of the parser. After completing these implementation tasks, you should run the parser, when it is configured to not produce output, and record how long it takes to parse five different XML documents. Finally, you and your partner should provide XML and output snippets and write a report summarizing your empirical results.

Summary of the Required Deliverables

You and your partner should always use a Git repository, hosted by Bitbucket, to store the source code, XML files, program output, and all of the other deliverables required by this assignment. The repository must be shared with the course instructor and the version control log should accurately reflect each student's contribution to the assignment. In addition, this assignment invites your partnership to submit one printed version of the following deliverables; each member should write and submit their own version of the first deliverable. Please see the instructor if you have any questions about the deliverables that you must turn in for this laboratory assignment.

1. A two paragraph commentary on the work that each team member completed.
2. A description of DOM-based XML parsing, examining its features, strengths, and weaknesses.
3. The final version of the `staff.xml` XML file that you customized and extended as appropriate.
4. The output from running the `ReadXMLFileWithDOM` program with both of the XML files.
5. The output from running the `DOMEcho` program with both of the provided XML files.
6. The source code of the `DOMEcho` program that you enhanced to produce less debugging output.
7. Snippets of the five XML files and the output from running the enhanced `DOMEcho` program.
8. A report explaining the performance trends associated with DOM-based XML parsing.

In adherence to the Honor Code, students should complete this assignment while exclusively collaborating with the other member of their team. While it is appropriate for students in this class—who are not in the same team—to have high-level conversations about the assignment, it is necessary to distinguish carefully between the team that discusses the principles underlying a problem with another team and the team that produces an assignment that is identical to, or merely a variation on, the work of another team. Deliverables from one team that are nearly identical to the work of another team will be taken as evidence of violating Allegheny College's Honor Code.