4.3 JSP

The first problem involved the Tomcat server crashing frequently. It turned out only happened when a new JavaBean class file had just been uploaded to the server and a page referencing the JavaBean was refreshed shortly after. If the page request reached the server during the second or so in which Tomcat noticed the new class file and began reloading it, the server would crash.

This was avoided by uploading class files and using the UNIX tail –f command to constantly monitor the Tomcat log file for the announcement that a newer version of the class file had been loaded. Only then was the page refreshed in the browser.

A miner, but annoying problem was found with the Java ResultSet.getString() method. This ResultSet[1] object stores rows that have been selected from a database table. The method to access column data for the currently selected row, getString() can be used by passing it either the column name as a String, or the column index as a primitive integer.

All collection classes in Java such as arrays and vectors use zero as the index of the first element. Therefore it would make sense to use getString(0) when referring to the first column of the ResultSet. For some reason though, this method uses 1 as the first column index which is inconsistant compared to the rest of the Java API and caused confusion.

Although not a problem as such, it was noticed that development seemed quite slow with JSP compared to experience of using PHP and ASP. This is mainly because changes to code need to be compiled for the JavaBeans, an extra step not necessary with PHP or ASP. Additionally, time was often spent waiting for Tomcat to notice the new JavaBean classes and reload them. Even though this happens within a couple of seconds, it was frustrating having to wait that long to see if a small bug fix or change has worked.

[1] Sun. 2002