

Logging with Log4J

Logging

All logging is done using Log4j (<http://logging.apache.org/log4j/>).

At the top of your class, define a static logger:

```
private static Logger log = Logger.getLogger(<classname>.class);
```

Then when putting in a log statement, you just do:

```
log.debug("This is the log message");
```

Please think about the log levels and use appropriately. Your code should have more than just log.debug statements! There is also info, warn, error, etc!

Also remember to use `isDebugEnabled()` appropriately. If your debug statement would take some effort to generate then stop the call occurring using `isDebugEnabled()`. But if the log message is very simple (say a concatenation of two strings), then it isn't worthwhile using `isDebugEnabled()`. This will help performance.

For example, assume you have an object `blah` that has a `toString` method that prints the values of 15 fields and contains some other complex objects, when method `validateBlah(Blah blah)` is called. If the log statement wants to dump out the object then do:

```
private boolean validateBlah(Blah blah) {  
    if ( log.isDebugEnabled() ) {  
        log.debug("--> validateBlah start. Blah is "+blah);  
    }  
    .....  
}
```

If all you are just logging the entry into the method don't bother with the `log.isDebugEnabled()` statement:

```
private boolean validateBlah(Blah blah) {  
    log.debug("--> validateBlah start.");  
    .....  
}
```