

Names and NSIDs: Wynston Ramsay - wcr723, Evan Snook - egs819, Darwin Zhang - ddz369
Project: jEdit
Change Number: 4
Date: 04/01/2018

Add timestamps to log

1. Change Request:

Locate where the activity.log is. Currently there are no timestamps in the log file. Add timestamps to all kinds of messages.

2. Concept Location:

We found that this change request was very intuitive to where the location would be. We used the solution explorer and looked up “log”. We managed to find the class that we needed to edit on the first try. We knew that there had to be one master log, so there was only really one file that we needed to find. Additionally, by observing the project’s structure, we were able to derive the likely location of the Log.java. Through previous experiences, it would make logical sense for it to be in the “util” package, as it is a utility, without a doubt.

Table 1. The list of all the classes visited during concept location.

#	File name	Tool used	Located?	Comments
1	org.gjt.sp.jedit /Log.java	Project Explorer	Changed	Found where the logging was done. Edited to add timestamps.
2.	org.gjt.sp.jedit /jEdit.java	Project Explorer	Unchanged	Wanted to check out the main file, to see how things were run. This was the first Change Request done, so it made sense at the time.

3. Impact Analysis:

Table 2. The list of all the classes visited during impact analysis.

#	Class name	Tool used	Impacted?	Comments
1.	org.gjt.sp.jedit/ Log.java	Eclipse Project Explorer	LogListModel, LogOutputStream	Found where the logging was done. Edited to add timestamps.

4. Learning process:

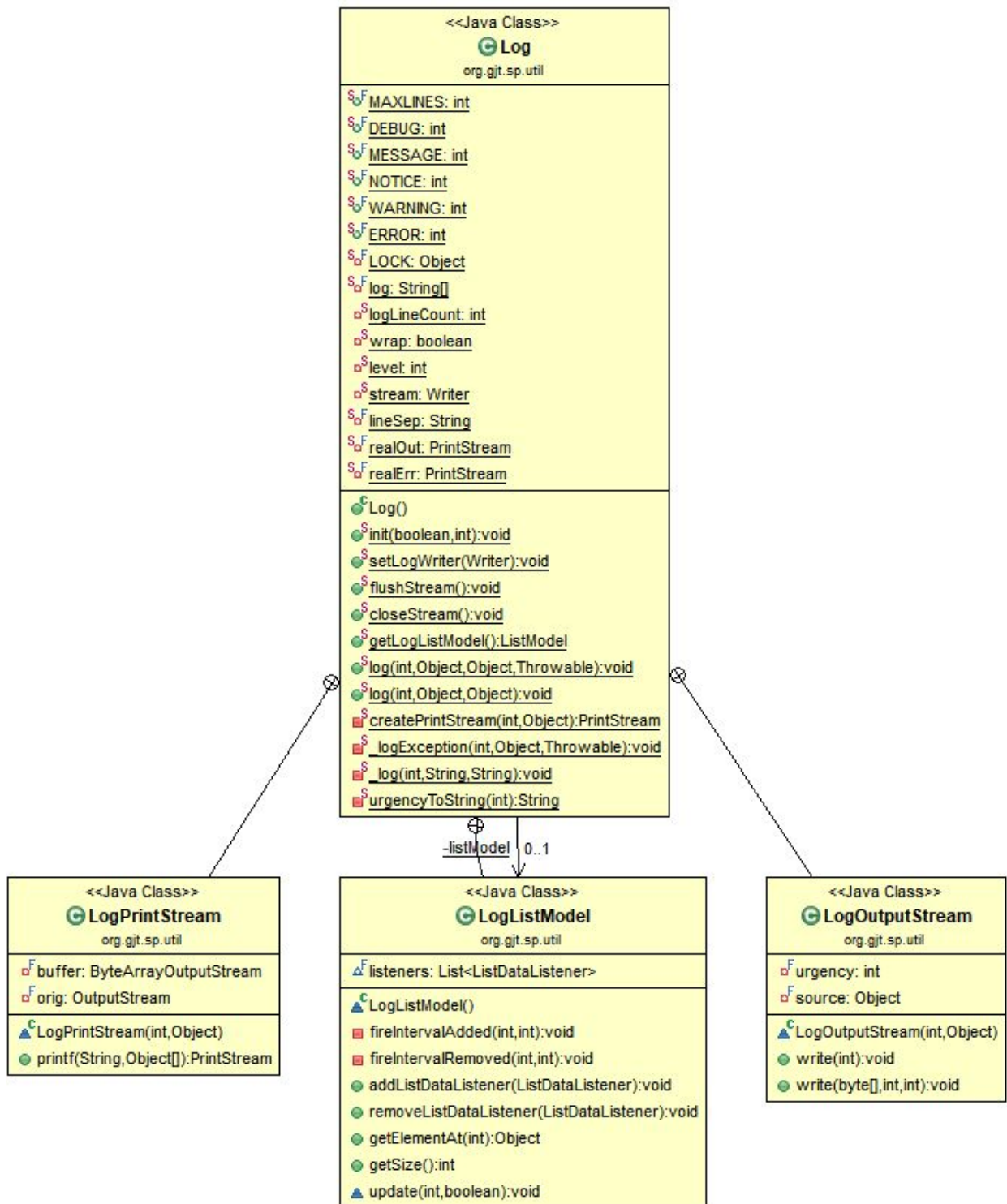


Figure 1: UML diagram of classes visited (generated by ObjectAid)

5. Description of the implementation:

The implementation was very straight forward. By checking out the log when running the program from the console, we were able to determine the structure of the log messages. It was fairly obvious that they were ran through Log.java, as there is no other file really like it. From there, we simply CTRL-F'd and skimmed through the class until we found the function that printed out the log strings, which we hence attached the timestamp to, to complete the change request.

6. Sources: jEdit, Eclipse, GitHub, ObjectAid UML Diagram (Eclipse Add-on)

7. Highlighted Source Code:

All modifications were made in Log.java
(See GitHub generated diff file below)

9 assignment 3/JEdit 4.3/org/gjt/sp/util/Log.java



@@ -26,6 +26,7 @@

```
26 26 import java.util.*;
27 27 import javax.swing.*;
28 28 import javax.swing.event.*;
```

29 +import java.sql.Timestamp; // #4

```
29 30
30 31 /**
31 32  * This class provides methods for logging events. In terms of functionality,
```



@@ -354,9 +355,11 @@ private static void _logException(final int urgency,

```
354 355 //{{{ _log() method
355 356 private static void _log(int urgency, String source, String message)
356 357 {
```

```
357 - String fullMessage = '[' + urgencyToString(urgency) + "]" + source
358 - + ": " + message;
359 -
```

```
358 + // #4 - changed log method to additionally log the timestamp
359 + Timestamp timeStamp = new Timestamp(System.currentTimeMillis());
360 + String fullMessage = timeStamp.toString() + ": [" + urgencyToString(urgency) + "]" +
361 + source + ": " + message;
362 +
```

```
360 363 try
361 364 {
362 365 log[logLineCount] = fullMessage;
```