Apache log4j™ 1.2

Welcome to Apache log4j, a logging library for Java. Apache log4j is an Apache Software Foundation Project and developed by a dedicated team of Committers of the Apache Software Foundation. For more info, please see The Apache Software Foundation . Apache log4j is also part of a project which is known as Apache Logging . Please see the License.

If you are interested in the recent changes, visit our changes report.

Why logging?

Inserting log statements into your code is a low-tech method for debugging it. It may also be the only way because debuggers are not always available or applicable. This is often the case for distributed applications.

On the other hand, some people argue that log statements pollute source code and decrease legibility. (We believe that the contrary is true). In the Java language where a preprocessor is not available, log statements increase the size of the code and reduce its speed, even when logging is turned off. Given that a reasonably sized application may contain thousands of log statements, speed is of particular importance.

Why log4j?

With log4j it is possible to enable logging at runtime without modifying the application binary. The log4j package is designed so that these statements can remain in shipped code without incurring a heavy performance cost. Logging behavior can be controlled by editing a configuration file, without touching the application binary.

Logging equips the developer with detailed context for application failures. On the other hand, testing provides quality assurance and confidence in the application. Logging and testing should not be confused. They are complementary. When logging is wisely used, it can prove to be an essential tool.

One of the distinctive features of log4j is the notion of inheritance in loggers. Using a logger hierarchy it is possible to control which log statements are output at arbitrarily fine granularity but also great ease. This helps to reduce the volume of logged output and the cost of logging.

The target of the log output can be a file, an OutputStream, a java.io.Writer, a remote log4j server, a remote Unix Syslog daemon, or many other output targets.

Performance

On an AMD Duron clocked at 800Mhz running JDK 1.3.1, it costs about 5 nanoseconds to determine if a logging statement should be logged or not. Actual logging is also quite fast, ranging from 21 microseconds using the SimpleLayout, 37 microseconds using the TTCCLayout. The performance of the PatternLayout is almost as good as the dedicated layouts, except that it is much more flexible.

Roadmap

The package is being constantly improved thanks to input from users and code contributed by authors in the community.

Please note, the team is currently working on log4j 2 which will replace log4j 1 in near future.