Apache: Log4j

return to the Apache Foundation

: **Note**: The following is an excerpt from the official Apache Log4j site. It is provided here as a connivence and is not authoritative. Refer to the original document as the authoritative reference.

Table 1: Data sheet for Log4j

Title	Log4j
Acronym	Log4j
Version	v. 2.11.2
Operating Systems	Java VM
Downloads	https://logging.apache.org/log4j/2.x/download.html
Guidelines	https://logging.apache.org/log4j/2.x/guidelines.html
Supported Languages	Java
Documentation	https://logging.apache.org/log4j/2.x/log4j-users-guide.pdf
License	Apache License, Version 2.0
Reference	https://logging.apache.org/log4j/2.x/index.html

Source: Log4J User's Guide |

Introduction

Almost every large application includes its own logging or tracing API. In conformance with this rule, the E.U. SEMPER project decided to write its own tracing API. This was in early 1996. After countless enhancements, several incarnations and much work that API has evolved to become log4j, a popular logging package for Java. The package is distributed under the Apache Software License, a fully-fledged open source license certified by the open source initiative. The latest log4j version, including full-source code, class files and documentation can be found at http://logging.apache.org/ log4j/2.x/index.html.

Inserting log statements into 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 usually the case for multithreaded applications and distributed applications at large.

Experience indicates that logging was an important component of the development cycle. It offers several advantages. It provides precise context about a run of the application. Once inserted into the code, the generation of logging output requires no human intervention. Moreover, log output can be saved in persistent medium to be studied at a later time. In addition to its use in the development cycle, a sufficiently rich logging package can also be viewed as an auditing tool.

> *As Brian W. Kernighan and Rob Pike put it in their truly excellent book "The Practice of Programming":*
>
> > *"As personal choice, we tend not to use debuggers beyond getting a stack trace or the value of a variable or two. One reason is that it is easy to get lost in details of complicated data structures and control flow; we find stepping through a program less productive than thinking harder and adding output statements and self-checking code at critical places. Clicking over statements takes longer than scanning the output of judiciously-placed displays. It takes less time to decide where to put print statements than to single-step to the critical section of code, even assuming we know where that is."*
>
> *More important, debugging statements stay with the program; debugging sessions are transient. Logging does have its drawbacks. It can slow down an application. If too verbose, it can cause scrolling blindness. To alleviate these concerns, log4j is designed to be reliable, fast and extensible. Since logging is rarely the main focus of an application, the log4j API strives to be simple to understand and to use.*

Features

Source: log4j Appenders

- Console, directly to the stdout or stderr stream.
- Console, using the PHP echo command.
- A file.
- A file (new file each day).
- A file (new file when a specified size has been reached).
- Sends the log via email. The entire log is sent in one email.
- Sends the log via email. Each log entry is sent in individual emails.
- MongoDB.
- Ignores all log events.
- Database.
- Creates a PHP user-level message using the PHP trigger_error() function.
- A network socket.
- Syslog

From:
https://omgwiki.org/dido/ - **DIDO Wiki**

Permanent link:
https://omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.b_stds:defact:apache:log4j&rev=1588726142

Last update: **2020/05/05 20:49**