

log4j是一个用Java编写的可靠，快速和灵活的日志框架（API），它在Apache软件许可下发布。Log4j已经被移植到了C，C++，C#，Perl，[Python](#)和Ruby等语言中。

Log4j是高度可配置的，并可通过在运行时的外部文件配置。它根据记录的优先级别，并提供机制，以指示记录信息到许多的目的地，诸如：数据库，文件，控制台，UNIX系统日志等。

Log4j中有三个主要组成部分：

- **loggers: 负责捕获记录信息。**
- **appenders : 负责发布日志信息，以不同的首选目的地。**
- **layouts: 负责格式化不同风格的日志信息。**

历史

- 始于1996年初的E.U. SEMPER（安全电子市场为欧洲）跟踪API的项目。
- 不计其数的改进，在几个大量的工作之后，API已经发展成为log4j，一个流行的日志记录包为Java。
- 该软件包是Apache软件许可证，由开源认证是一个不折不扣的开源许可证下发布。
- 最新log4j的版本，包括完整的源代码，类文件和文档可以在这里找到 <http://logging.apache.org/log4j/>.

特性：

- log4j的是线程安全的
- log4j是经过优化速度的
- log4j是基于一个名为记录器的层次结构
- log4j的支持每个记录器多输出追加器（appender）
- log4j支持国际化。
- log4j并不限于一组预定义的设备
- 日志行为可以使用配置文件在运行时设置
- log4j设计从一开始就是处理Java异常
- log4j使用多个层次，即ALL，TRACE，DEBUG，INFO，WARN，ERROR和FATAL
- 日志输出的格式可以通过扩展Layout类容易地改变
- 日志输出的目标，以及在写入策略可通过实现Appender程序接口改变
- log4j 会故障停止。然而，尽管它肯定努力确保传递，log4j不保证每个日志语句将被传递到目的地。

Log4j API设计为分层结构，其中每一层提供了不同的对象，对象执行不同的任务。这使得设计灵活，根据将来需要来扩展。

有两种类型可用在Log4j的框架对象。

- **核心对象： 框架的强制对象和框架的使用。**
- **支持对象： 框架和支持体核心对象，可选的对象执行另外重要的任务。**

核心对象：

Logger对象:

顶级层的Logger，它提供Logger对象。Logger对象负责捕获日志信息及它们存储在一个空间的层次结构。

布局对象:

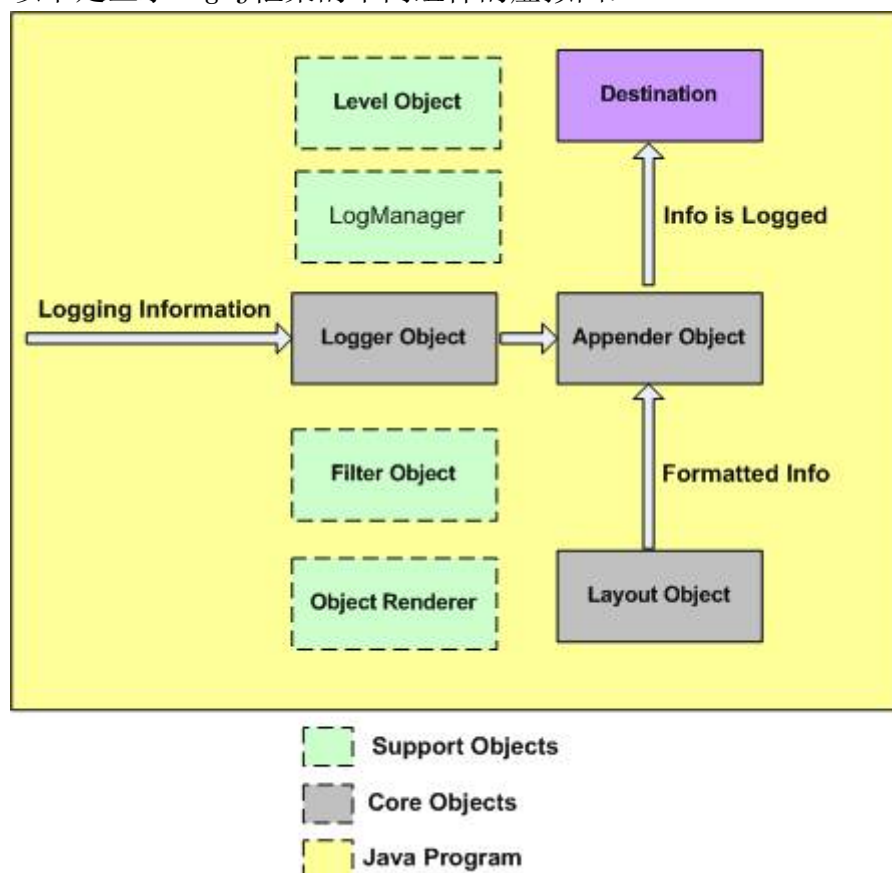
该层提供其用于格式化不同风格的日志信息的对象。布局层提供支持Appender对象到发布日志信息之前。

布局对象的发布方式是人类可读的及可重复使用的记录信息的一个重要的角色。

Appender对象:

下位层提供Appender对象。Appender对象负责发布日志信息，以不同的首选目的地，如数据库，文件，控制台，UNIX系统日志等。

以下是显示Log4J框架的不同组件的虚拟图:



支持对象:

log4j框架的其他重要的对象起到日志框架的一个重要作用:

Level对象:

级别对象定义的任何记录信息的粒度和优先级。有记录的七个级别在API中定义: OFF, DEBUG, INFO, ERROR, WARN, FATAL 和 ALL

Filter对象:

过滤对象用于分析日志信息及是否应记录或不用这些信息做出进一步的决定。

一个appender对象可以有与之关联的几个Filter对象。如果日志记录信息传递给特定Appender对象，都和特定Appender相关的Filter对象批准的日志信息，然后才能发布到所连接的目的地。

对象渲染器:

ObjectRenderer对象是一个指定提供传递到日志框架的不同对象的字符串表示。这个对象所使用的布局对象来准备最后的日志信息。

日志管理：

日志管理对象管理的日志框架。它负责从一个系统级的配置文件或配置类读取初始配置参数。