Log4j Configuration Guide

Table of Contents

1. Introduction.....

Chapter 1. Introduction

This is very short explanation of log4j configuration rules. Detailed specification can be found at: http://jakarta.apache.org/log4j/docs/documentation.html [http://jakarta.apache.org/log4j/docs/documentation.html].

Log4j has three main components: loggers, appenders and layouts. Loggers are named entities (com.foo is a parent of the logger named com.foo.Bar). The root logger can be configured with (log4j.xml):

```
<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<log4j:configuration xmlns:log4j='http://jakarta.apache.org/log4j/'>

    <appender name="ROLL" class="org.apache.log4j.RollingFileAppender">

        <param name="File" value="C:/enhydra5.1/logs/multiserver.log"/>

        <param name="MaxFileSize" value="10MB"/>

        <param name="MaxBackupIndex" value="2"/>

        <layout class="org.apache.log4j.PatternLayout">

            <param name="ConversionPattern" value="%d{ISO8601}: [%t] %C{1}, %p, %c: %m%n"/>

        </layout>

    </appender>

    <root>

        <level value ="info"/>

        <appender-ref ref="ROLL"/>

    </root>

</log4j:configuration>
```

where <level> is one of the: ALL, DEBUG, INFO, WARN, ERROR, FATAL, OFF.

ALL includes all messages, OFF does not show any message, DEBUG is for verbose output and with FATAL only critical errors should be logged.

Log4j also allows logging requests to print to multiple output destinations called appenders (<appender-ref> in example above). One of the distinctive features of log4j is the notion of inheritance in loggers. For example, the output of a log statement of logger C will go to all the appenders in C and its ancestors. However, if an ancestor of logger C, say P, has the additivity flag set to false, then C's output will be directed to all the appenders in C and it's ancestors upto and including P but not the appenders in any of the ancestors of P. Loggers have their additivity flag set to true by default.

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 even a NT Event logger among many other output targets.

Hence, appender type is mostly one of: ConsoleAppender, FileAppender, RollingFileAppender, DailyRollingFileAppender, SocketAppender, JMSAppender, SMTPAppender, AsyncAppender.

If you want to customize not only the output destination but also the output format, it can be accomplished by associating a layout with an appender. For example, the PatternLayout with the conversion pattern %r [%t]%-5p %c - %m%n will output something akin to:

```
176 [main] INFO  org.foo.Bar - Located nearest gas station.
```

In the output above: the first field equals the number of milliseconds elapsed since the start of the program, the second field indicates the thread making the log request, the third field represents the priority of the log statement, the fourth field equals the name of the logger associated with the log request, the text after the - indicates the statement's message.

That's all for this very short guide.