

Log4j配置详解之log4j.xml - IT民工

Log4J的配置文件(Configuration File)就是用来设置记录器的级别、存放器和布局的，它可接key=value格式的设置或xml格式的设置信息。通过配置，可以创建出Log4J的运行环境。

一、log4j.jar包下载地址：<http://logging.apache.org/log4j/1.2/download.html>

二、配置文件类型：log4j配置文件可以是log4j.xml也可以是log4j.properties

log4j.xml配置说明

1. <?xml version="1.0" encoding="UTF-8"?>
2. <!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
3. <log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/" >
4. <appender name="myConsole" class="org.apache.log4j.ConsoleAppender">
5. <layout class="org.apache.log4j.PatternLayout">
6. <param name="ConversionPattern"
7. value="[%d{dd HH:mm:ss,SSS} %-5p] [%t] %c{2} - %m%n" />
8. </layout>
9. <!--过滤器设置输出的级别-->
10. <filter class="org.apache.log4j.varia.LevelRangeFilter">
11. <param name="levelMin" value="debug" />
12. <param name="levelMax" value="warn" />
13. <param name="AcceptOnMatch" value="true" />
14. </filter>
15. </appender>
16. <appender name="myFile" class="org.apache.log4j.RollingFileAppender">
17. <param name="File" value="D:/output.log" /><!-- 设置日志输出文件名 -->
18. <!-- 设置是否在重新启动服务时，在原有日志的基础添加新日志 -->
19. <param name="Append" value="true" />
20. <param name="MaxBackupIndex" value="10" />
21. <layout class="org.apache.log4j.PatternLayout">
22. <param name="ConversionPattern" value="%p (%c:%L)- %m%n" />
23. </layout>
24. </appender>
25. <appender name="activexAppender" class="org.apache.log4j.DailyRollingFileAppender">
26. <param name="File" value="E:/activex.log" />
27. <param name="DatePattern" value="'.yyyy-MM-dd'.log' />

```

28.     <layout class="org.apache.log4j.PatternLayout">
29.         <param name="ConversionPattern"
30.             value="[%d{MMdd HH:mm:ss SSS}\ %-5p] [%t] %c{3} - %m%n" />
31.     </layout>
32. </appender>
33. <!-- 指定logger的设置，additivity指示是否遵循缺省的继承机制-->
34. <logger name="com.runway.bssp.activeXdemo" additivity="false">
35.     <priority value ="info"/>
36.     <appender-ref ref="activexAppender" />
37. </logger>
38. <!-- 根logger的设置-->
39. <root>
40.     <priority value ="debug"/>
41.     <appender-ref ref="myConsole"/>
42.     <appender-ref ref="myFile"/>
43. </root>
44. </log4j:configuration>

```

(1). 输出方式appender一般有5种：

org.apache.log4j.RollingFileAppender(滚动文件，自动记录最新日志)

org.apache.log4j.ConsoleAppender (控制台)

org.apache.log4j.FileAppender (文件)

org.apache.log4j.DailyRollingFileAppender (每天产生一个日志文件)

org.apache.log4j.WriterAppender (将日志信息以流格式发送到任意指定的地方)

(2). 日记记录的优先级priority，优先级由高到低分为

OFF ,FATAL ,ERROR ,WARN ,INFO ,DEBUG ,ALL。

Log4j建议只使用FATAL ,ERROR ,WARN ,INFO ,DEBUG这五个级别。

(3). 格式说明layout中的参数都以%开始，后面不同的参数代表不同的格式化信息（参数按字母表顺序列出）：

%c 输出所属类的全名，可在修改为 %d{Num} ,Num类名输出的维
（如："org.apache.elathen.ClassName",%C{2}将输出elathen.ClassName）

%d 输出日志时间其格式为 %d{yyyy-MM-dd HH:mm:ss,SSS}，可指定格式 如 %d{HH:mm:ss}

%l 输出日志事件发生位置，包括类目名、发生线程，在代码中的行数

%n 换行符

%m 输出代码指定信息，如info("message"),输出message

%p 输出优先级，即 FATAL ,ERROR 等

%r 输出从启动到显示该log信息所耗费的毫秒数

%t 输出产生该日志事件的线程名

log4j.xml配置文件节点详解

xml declaration and DTD

xml配置文件的头部包括两个部分：xml声明和DTD声明。头部的格式如下：

Xml代码



1. <?xml version="1.0" encoding="UTF-8" ?>
2. <!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

log4j:configuration (root element)

- xmlns:log4j [#FIXED attribute]：定义log4j的名字空间，取定值"http://jakarta.apache.org/log4j/"
- appender [* child]：一个appender子元素定义一个日志输出目的地
- logger [* child]：一个logger子元素定义一个日志写出器
- root [? child]：root子元素定义了root logger

appender

appender元素定义一个日志输出目的地。

- name [#REQUIRED attribute]：定义appender的名字，以便被后文引用
- class [#REQUIRED attribute]：定义appender对象所属的类的全名
- param [* child]：创建appender对象时传递给类构造方法的参数
- layout [? child]：该appender使用的layout对象

layout

layout元素定义与某一个appender相联系的日志格式化器。

- class [#REQUIRED attribute] : 定义layout对象所属的类的全名
- param [* child] : 创建layout对象时传递给类构造方法的参数

logger

logger元素定义一个日志输出器。

- name [#REQUIRED attribute] : 定义logger的名字，以便被后文引用
- additivity [#ENUM attribute] : 取值为"true"（默认）或者"false"，是否继承父logger的属性
- level [? child] : 定义该logger的日志级别
- appender-ref [* child] : 定义该logger的输出目的地

root

root元素定义根日志输出器root logger。

- param [* child] : 创建root logger对象时传递给类构造方法的参数
- level [? child] : 定义root logger的日志级别
- appender-ref [* child] : 定义root logger的输出目的地

level

level元素定义logger对象的日志级别。

- class [#IMPLIED attribute] : 定义level对象所属的类，默认情况下是"org.apache.log4j.Level类
- value [#REQUIRED attribute] : 为level对象赋值。可能的取值从小到大依次为"all"、"debug"、"info"、"warn"、"error"、"fatal"和"off"。当值为"off"时表示没有任何日志信息被输出
- param [* child] : 创建level对象时传递给类构造方法的参数

appender-ref

appender-ref元素引用一个appender元素的名字，为logger对象增加一个appender。

- ref [#REQUIRED attribute] : 一个appender元素的名字的引用
- appender-ref元素没有子元素

param

param元素在创建对象时为类的构造方法提供参数。它可以成为appender、layout、filter、errorHandler、level、categoryFactory和root等元素的子元素。

- name and value [#REQUIRED attributes] : 提供参数的一组名值对
- param元素没有子元素

在xml文件中配置appender和layout

创建不同的Appender对象或者不同的Layout对象要调用不同的构造方法。可以使用param子元素来设定不同的参数值。

创建ConsoleAppender对象

ConsoleAppender的构造方法不接受其它的参数。

Xml代码



1.
2. <appender name="console.log" class="org.apache.log4j.ConsoleAppender">
3. <layout ... >
4.
5. </layout>
6. </appender>
7.

创建FileAppender对象

可以为FileAppender类的构造方法传递两个参数：File表示日志文件名；Append表示如文件已存在，是否把日志追加到文件尾部，可能取值为"true"和"false"（默认）。

Xml代码



1.
2. <appender name="file.log" class="org.apache.log4j.FileAppender">
3. <param name="File" value="/tmp/log.txt" />
4. <param name="Append" value="false" />
5. <layout ... >
6.
7. </layout>
8. </appender>
9.

创建RollingFileAppender对象

除了File和Append以外，还可以为RollingFileAppender类的构造方法传递两个参数：
MaxBackupIndex备份日志文件的个数（默认是1个）；MaxFileSize表示日志文件允许的最大字节数（默认是10M）。

Xml代码



1.
2. <appender name="rollingFile.log" class="org.apache.log4j.RollingFileAppender">
3. <param name="File" value="/tmp/rollingLog.txt" />
4. <param name="Append" value="false" />
5. <param name="MaxBackupIndex" value="2" />
6. <param name="MaxFileSize" value="1024" />
7. <layout ... >
8.
9. </layout>
10. </appender>
11.

Xml代码



1.
2. <layout class="org.apache.log4j.PatternLayout">
3. <param name="Conversion" value="%d [%t] %p - %m%n" />
4. </layout>
5.