WEBLOGIC配置LOG4J

2009.09.06

王凡

Wf141732@sohu.com

woshiwangfan@gmail.com

1.1. 配置

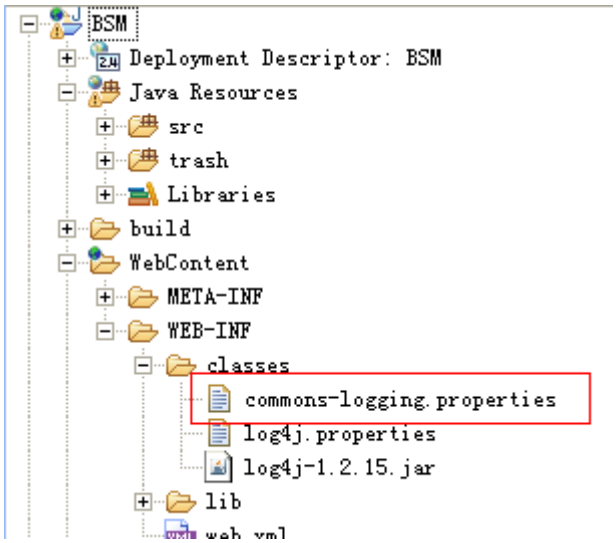
在tomcat的项目中中配置log4j很简单，将log4j的jar加到app中就可以了。

但是在weblogic的项目中却又不一样。

1.1.1. Common-logging的配置

Log4j与commons-logging的类包weblogic已经默认提供。

在workshop为新建的app创建了一个Common-logging，在项目目录下面WebContent\WEB-INF\classes\commons-loggins.properties



打开commons-loggins.properties可以看到

```
org.apache.commons.logging.LogFactory=weblogic.logging.commons.
LogFactoryImpl
```

weblogic的log文件换成log4j

```
org.apache.commons.logging.Log=org.apache.commons.logging.impl.
Log4JLogger
```

这个在tomacat平台是没有问题，但是在weblogic中会导致项目不能发布。

```
org.apache.commons.logging.LogFactory =
org.apache.commons.logging.impl.Log4JCategoryLog
```

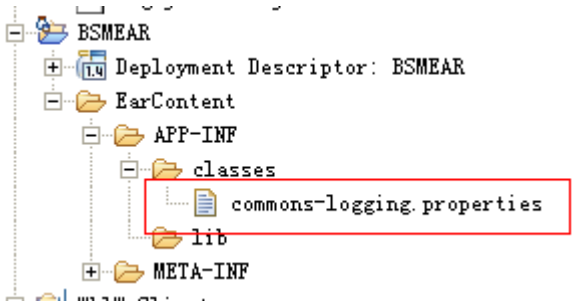
或者将已经存在的注释掉，commons-logging会自己进行选择。具体的选择顺序可google一下，这里不做赘述。

此时再次运行程序还是报下面的错。

```
Exception in thread "Main Thread"
java.lang.ExceptionInInitializerError
Caused by:
org.apache.commons.logging.LogConfigurationException: The
chosen LogFactory implementation does not extend LogFactory.
Please check your configuration. .....
```

这是因为在我们建项目的时候建了一个EAR，这个是作为发布的。

在这个EAR项目中找到commons-loggins.properties



还是将配置文件换为log4j。

```
org.apache.commons.logging.Log=org.apache.commons.logging.impl.
Log4JLogger
```

在这里我们用到了commons-logging这个jar包，这个包的作用是作为一个接口作用，用它可以和很多的其它的log进行关联，而且可以使项目对单个的log依赖过大。

可以对commons-logging进行配置来换其它的log

```
#Weblogic
#org.apache.commons.logging.LogFactory=weblogic.logging.commons
.LogFactoryImpl

#LogFactory implement
#org.apache.commons.logging.LogFactory=org.apache.commons.loggi
ng.impl.LogFactoryImpl

# SimpleLog
#org.apache.commons.logging.Log =
org.apache.commons.logging.impl.SimpleLog

# JDK 1.4 logger##这个在jdk1.4以上都可以用
#org.apache.commons.logging.Log=org.apache.commons.logging.impl
.Jdk14Logger

# Avalon Toolkit
#org.apache.commons.logging.Log=org.apache.commons.logging.impl
.LogKitLogger

#Log4j
#org.apache.commons.logging.Log=org.apache.commons.logging.impl
.Log4JLogger
```

1.1.2. 使用log4j

在应用程序中使用log4j相当简单，直接添加commons-logging的两个引用。

```
import org.apache.commons.logging.LogFactory;
import org.apache.commons.logging.Log;
```

定义log

```
private static Log logger = LogFactory.getLog(test.class);
```

然后就可以使用了。

完整的代码如下

```
package cn.com.tsingtao.logistics.service;

import javax.xml.rpc.server.ServiceLifecycle;

import org.apache.commons.logging.LogFactory;

import org.apache.commons.logging.Log;

public class test {

   private static Log logger = LogFactory.getLog(test.class);

   public static void main(String[] args) {

            logger.warn("--------------lallal----------------
");
```

```
            logger.error("-------------lallal----------------
");
    }


}
```

也可以不使用commons-logging直接使用log4j

代码如下

```
package cn.com.tsingtao.logistics.service;

import javax.xml.rpc.server.ServiceLifecycle;

import org.apache.log4j.Logger;

import org.apache.log4j.PropertyConfigurator;

public class test {

   private static Logger logger =
Logger.getLogger(test.class );

   public static void main(String[] args) {

            logger.warn("---------lallal-----------------");

            logger.error("---------lallal-----------------");

   }


}
```

红色部分是与使用了接口commons-logging的log4j不同的地方。

1.1.3. 配置简单log4j

Log4j的配置主要有两个方面的内容：

配置根logger，主要是定义输出的信息类型和输出目的地，在这里INFO是指高于INFO级别的信息都需要输出包括INFO,WARN,ERRO,FATAL。然后是定义输出目标，这里有三个，只是三个名字，并没有定义具体的目的介质。

```
log4j.rootLogger=INFO, CONSOLE, ALLINFO, ERRORINFO
```

然后是具体配置单个的目的，首先定义的CONSOLE，它可以输出的信息最低级别为INFO，其次是定义其输出类型，这里定义的是console，即控制台输出。再个是输出信息布局，最后是单条信息的格式。

```
og4j.appender.CONSOLE.Threshold=INFO##定义信息级别
log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
```

```
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern=%d{yyyy-MM-dd
HH:mm:ss} <%c>[%t] %-5p : %m%n
```

1.1.4. Log4j的输出类型

在根里面我们定义了三个输出目的地，有三个输出类型。Log4j常用的可以定义四种输出类型

```
log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
log4j.appender.ALLINFO=org.apache.log4j.DailyRollingFileAppende
r
log4j.appender.file=org.apache.log4j.FileAppender
log4j.appender.MF=org.apache.log4j.RollingFileAppender
```

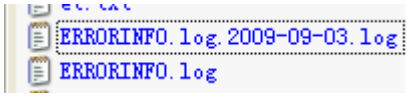
分别对应的是控制台，每日日志归档，单个日志文件，根据日志文件大小归档。

控制台的上面已经有一个例子，其它的分别写一个配置。

- 每日日志归档

```
log4j.appender.ERRORINFO.Threshold=ERROR
log4j.appender.ERRORINFO=org.apache.log4j.DailyRollingFileAppen
der
log4j.appender.ERRORINFO.File=H:\\tmp\\ERRORINFO.log
log4j.appender.ERRORINFO.DatePattern='.'yyyy-MM-dd'.log'
log4j.appender.ERRORINFO.layout=org.apache.log4j.PatternLayout
log4j.appender.ERRORINFO.layout.ConversionPattern=%d{yyyy-MM-dd
HH:mm:ss} <%c>[%F:%L] %-5p : %m%n
```

在这个文件中只记录错误的信息，文件放到ERRORINFO.log中，这个文件会在第二天由系统重新命名为ERRORINFO.log.2009-09-03.log。



当天的文件依旧会记在ERRORINFO.log中，重新命名的规则则是根据DatePattern这个参数在设置。

- 单个日志文件

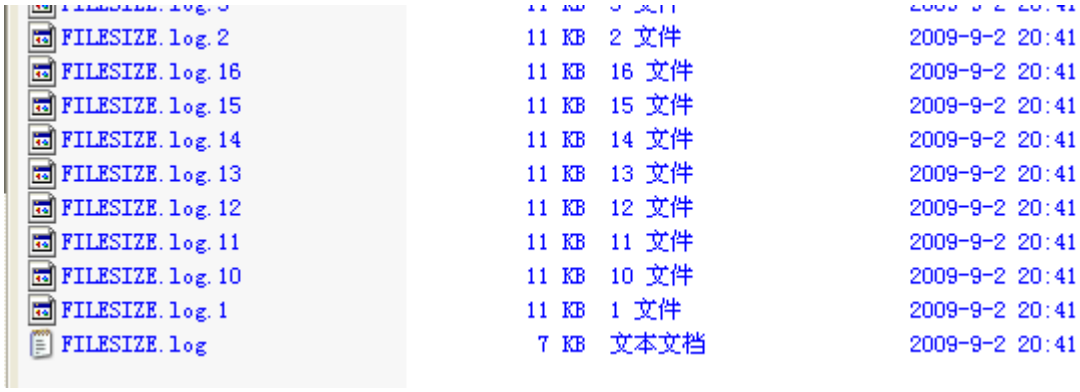
单个日志文件很简单

```
log4j.appender.ERRORINFO.Threshold=ERROR
log4j.appender.ERRORINFO=org.apache.log4j. FileAppender
log4j.appender.ERRORINFO.File=H:\\tmp\\ERRORINFO.log
log4j.appender.ERRORINFO.layout=org.apache.log4j.PatternLayout
log4j.appender.ERRORINFO.layout.ConversionPattern=%d{yyyy-MM-dd
HH:mm:ss} <%c>[%F:%L] %-5p : %m%n
```

- 根据日志文件大小归档

```
log4j.appender.FILESIZE=org.apache.log4j.RollingFileAppender
log4j.appender.FILESIZE.File=H:\\tmp\\FILESIZE.log
log4j.appender.FILESIZE.maxBackupIndex=1000
log4j.appender.FILESIZE.maxFileSize=10KB
log4j.appender.FILESIZE.layout=org.apache.log4j.PatternLayout
log4j.appender.FILESIZE.layout.ConversionPattern=%d{yyyy-MM-dd
HH:mm:ss} <%c>[%F:%L] %-5p : %m%n
```

当文件大小冲过1M后产生新的文件，命名是根据原命名加序号,这里有个最大序号，当产生的文件超过最大序号后，将覆盖以前产生的文件。



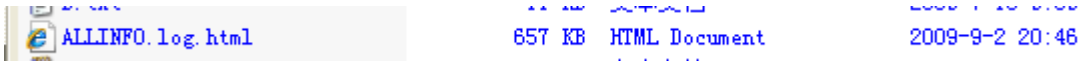
1.1.5. log4j的布局类型

信息布局有四种，分别是：

```
org.apache.log4j.HTMLLayout（以HTML表格形式布局），
org.apache.log4j.PatternLayout（可以灵活地指定布局模式），
org.apache.log4j.SimpleLayout（包含日志信息的级别和信息字符串），
org.apache.log4j.TTCCLayout（包含日志产生的时间、线程、类别等等信息）
```

- 以HTML表格形式布局

```
log4j.appender.ALLINFO.Threshold=INFO
log4j.appender.ALLINFO=org.apache.log4j.DailyRollingFileAppende
r
log4j.appender.ALLINFO.File=H:\\tmp\\ALLINFO.log.html
log4j.appender.ALLINFO.DatePattern='.'yyyy-MM-dd'.html'
log4j.appender.ALLINFO.layout=org.apache.log4j.HTMLLayout
log4j.appender.ALLINFO.layout.ConversionPattern=%d{yyyy-MM-dd
HH:mm:ss} <%c>[%F:%L] %-5p : %m%n
```



Log session start time Wed Sep 02 20:46:22 CST 2009

Time	Thread	Level	Category	Message
15	Main Thread	WARN	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------0
31	Main Thread	ERROR	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------0
31	Main Thread	WARN	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------1
31	Main Thread	ERROR	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------1
94	Main Thread	WARN	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------2
94	Main Thread	ERROR	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------2
94	Main Thread	WARN	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------3
94	Main Thread	ERROR	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------3
94	Main Thread	WARN	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------4
94	Main Thread	ERROR	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------4
94	Main Thread	WARN	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------5
94	Main Thread	ERROR	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------5
94	Main Thread	WARN	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------6
94	Main Thread	ERROR	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------6
94	Main Thread	WARN	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------7
94	Main Thread	ERROR	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------7
94	Main Thread	WARN	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------8
94	Main Thread	ERROR	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------8
94	Main Thread	WARN	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------9
94	Main Thread	ERROR	cn.com.tsingtao.logistics.service.test	-------------------lallal----------------------9

- 灵活地指定布局模式

```
2009-09-02 20:46:25 <cn.com.tsingtao.logistics.service.test>[Main Thread] WARN  : --------
2009-09-02 20:46:25 <cn.com.tsingtao.logistics.service.test>[Main Thread] ERROR : --------
2009-09-02 20:46:25 <cn.com.tsingtao.logistics.service.test>[Main Thread] WARN  : --------
2009-09-02 20:46:25 <cn.com.tsingtao.logistics.service.test>[Main Thread] ERROR : --------
2009-09-02 20:46:25 <cn.com.tsingtao.logistics.service.test>[Main Thread] WARN  : --------
2009-09-02 20:46:25 <cn.com.tsingtao.logistics.service.test>[Main Thread] ERROR : --------
2009-09-02 20:46:25 <cn.com.tsingtao.logistics.service.test>[Main Thread] WARN  : --------
2009-09-02 20:46:25 <cn.com.tsingtao.logistics.service.test>[Main Thread] ERROR : --------
2009-09-02 20:46:25 <cn.com.tsingtao.logistics.service.test>[Main Thread] WARN  : --------
2009-09-02 20:46:25 <cn.com.tsingtao.logistics.service.test>[Main Thread] ERROR : --------
```

- 简单的布局模式

```
ERROR - -------------------lallal----------------------995
WARN - -------------------lallal----------------------996
ERROR - -------------------lallal----------------------996
WARN - -------------------lallal----------------------997
ERROR - -------------------lallal----------------------997
WARN - -------------------lallal----------------------998
ERROR - -------------------lallal----------------------998
WARN - -------------------lallal----------------------999
ERROR - -------------------lallal----------------------999
```

➢ 多信息布局模式

```
[Main Thread] ERROR cn.com.tsingtao.logistics.service.test - -------------------lallal------
[Main Thread] WARN cn.com.tsingtao.logistics.service.test - -------------------lallal-------
[Main Thread] ERROR cn.com.tsingtao.logistics.service.test - -------------------lallal------
[Main Thread] WARN cn.com.tsingtao.logistics.service.test - -------------------lallal-------
[Main Thread] ERROR cn.com.tsingtao.logistics.service.test - -------------------lallal------
[Main Thread] WARN cn.com.tsingtao.logistics.service.test - -------------------lallal-------
[Main Thread] ERROR cn.com.tsingtao.logistics.service.test - -------------------lallal------
[Main Thread] WARN cn.com.tsingtao.logistics.service.test - -------------------lallal-------
[Main Thread] ERROR cn.com.tsingtao.logistics.service.test - -------------------lallal------
```

➢ 指定布局

```
2009-09-02 21:02:56 <cn.com.tsingtao.logistics.service.test>[Main Thread] ERROR : --------------
2009-09-02 21:02:56 <cn.com.tsingtao.logistics.service.test>[Main Thread] WARN  : --------------
2009-09-02 21:02:56 <cn.com.tsingtao.logistics.service.test>[Main Thread] ERROR : --------------
2009-09-02 21:02:56 <cn.com.tsingtao.logistics.service.test>[Main Thread] WARN  : --------------
2009-09-02 21:02:56 <cn.com.tsingtao.logistics.service.test>[Main Thread] ERROR : --------------
2009-09-02 21:02:56 <cn.com.tsingtao.logistics.service.test>[Main Thread] WARN  : --------------
2009-09-02 21:02:56 <cn.com.tsingtao.logistics.service.test>[Main Thread] ERROR : --------------
2009-09-02 21:02:56 <cn.com.tsingtao.logistics.service.test>[Main Thread] WARN  : --------------
2009-09-02 21:02:56 <cn.com.tsingtao.logistics.service.test>[Main Thread] ERROR : --------------
```

指定布局的输出信息比其它模式下都要多，因为在这个模式下面我们可以指定输出信息。

1.1.6. 自定义日志输出内容

在每个定义输出的模块后面都是定义单条信息的输出样式和信息的

```
log4j.appender.ALLINFO.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} <%c>[%F:%L] %-5p : %m%n
```

%p: 输出日志信息优先级，即DEBUG，INFO，WARN，ERROR，FATAL,

%d: 输出日志时间点的日期或时间，默认格式为ISO8601，也可以在其后指定格式，比如：%d{ yyyy-MM-dd HH:mm:ss,sss}，输出类似：2009-09-02 21:08:05,005

%r: 输出自应用启动到输出该log信息耗费的毫秒数

%c: 输出日志信息所属的类目，通常就是所在类的全名，比如这里的 cn.com.tsingtao.logistics.service.test

%t: 输出产生该日志事件的线程名

%l: 输出日志事件的发生位置，相当于%C.%M[%F:%L]的组合,包括类目名、发生的线程，以及在代码中的行数。举例：Main Thread [test.java:73]

%x: 输出和当前线程相关联的NDC(嵌套诊断环境),尤其用到像java servlets这样的多客户多线程的应用中。

%%: 输出一个"%"字符

%F: 输出日志消息产生时所在的文件名称

%L: 输出代码中的行号

%m: 输出代码中指定的消息,产生的日志具体信息

%n: 输出一个回车换行符，Windows平台为"\r\n"，Unix平台为"\n"输出日志信息换行

可以在%与模式字符之间加上修饰符来控制其最小宽度、最大宽度、和文本的对齐方式。如：

1)%20c：指定输出category的名称，最小的宽度是20，如果category的名称小于20的话，默认的情况下右对齐。

2)%-20c:指定输出category的名称，最小的宽度是20，如果category的名称小于20的话，"-"号指定左对齐。

3)%.30c:指定输出category的名称，最大的宽度是30，如果category的名称大于30的话，就会将左边多出的字符截掉，但小于30的话也不会有空格。

4)%20.30c:如果category的名称小于20就补空格，并且右对齐，如果其名称长于30字符，就从左边较远输出的字符截掉。

```
2009-09-02 17:00:40 <cn.com.tsingtao.logistics.service.test>[test.java:39
2009-09-02 17:00:40 <cn.com.tsingtao.logistics.service.test>[test.java:39
2009-09-02 17:00:40 <cn.com.tsingtao.logistics.service.test>[test.java:39
2009-09-02 17:00:40 <cn.com.tsingtao.logistics.service.test>[test.java:39
2009-09-02 17:00:40 <cn.com.tsingtao.logistics.service.test>[test.java:39
2009-09-02 17:00:40 <cn.com.tsingtao.logistics.service.test>[test.java:39
2009-09-02 17:00:40 <cn.com.tsingtao.logistics.service.test>[test.java:39
2009-09-02 17:00:40 <cn.com.tsingtao.logistics.service.test>[test.java:39
2009-09-02 17:00:40 <cn.com.tsingtao.logistics.service.test>[test.java:39
2009-09-02 17:00:40 <cn.com.tsingtao.logistics.service.test>[test.java:39
```

1.1.7. log4j日志文件分类输出

在项目上，可能客户会只想看错误信息，那么我们就要专门的将错误的日志信息单独的放在一个文件中。在上面的输出中我们第一句都有这样一句

```
log4j.appender.ERRORINFO.Threshold=ERROR
```

这个就是只将错误，或者比错误信息更严重的信息输出。

然后可以再定义个一个文件将一般的信息输出，这样在查看错误信息后就可以根据错误信息到记录更全的一般信息的日志文件中查找问题。

但是有的客户想要将各个信息单独存放在单独的日志文件中，INFO里面只能放INFO信息，ERROR里面只能放ERROR信息。这个时候properties已经不再支持了。在一般的java开源项目中都支持properties和xml两种文件配置模式。Apach也提供了xml的配置，而且两种模式的支持也不一样，分文件存放只在xml配置文件中支持。

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<!--#log4j中有5级logger ,#FATAL 0 ,#ERROR 3 ,#WARN 4 ,#INFO
6 ,#DEBUG 7 -->
<log4j:configuration
xmlns:log4j='http://jakarta.apache.org/log4j/'>

   <!--输出到控制台-->
   <!--
            <appender name="LOG.Console"
class="org.apache.log4j.ConsoleAppender">
            <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%d{yyy-MM-dd
HH:mm:ss} [%-5p] %c {%F:%L} - %m%n" />
            </layout>
            </appender>
   -->

   <!--将级别为DEBUG的信息输出到控制台-->
   <appender name="LOG.DEBUG"
            class="org.apache.log4j.RollingFileAppender">
            <param name="File" value="d:/log/debug.log" />
            <param name="MaxFileSize" value="5120KB" />
            <param name="MaxBackupIndex" value="10" />
            <layout class="org.apache.log4j.PatternLayout">
                     <param name="ConversionPattern"
                              value="%d{yyy-MM-dd HH:mm:ss} [%-
5p] %c {%F:%L} - %m%n" />
            </layout>
            <filter
class="org.apache.log4j.varia.LevelRangeFilter">
                     <param name="LevelMin" value="DEBUG" />
                     <param name="LevelMax" value="DEBUG" />
            </filter>
   </appender>

   <!--将级别为INFO的信息输出到控制台-->
   <appender name="LOG.INFO"
            class="org.apache.log4j.RollingFileAppender">
            <param name="File" value="d:/log/info.log" />
            <param name="MaxFileSize" value="5120KB" />
            <param name="MaxBackupIndex" value="10" />
            <layout class="org.apache.log4j.PatternLayout">
                     <param name="ConversionPattern"
                              value="%d{yyy-MM-dd HH:mm:ss} [%-
5p] %c {%F:%L} - %m%n" />
            </layout>
            <filter
class="org.apache.log4j.varia.LevelRangeFilter">
                     <param name="LevelMin" value="INFO" />
                     <param name="LevelMax" value="INFO" />
            </filter>
   </appender>
```

```
    <!--将级别为WARN的信息输出到控制台-->
    <appender name="LOG.WARN"
            class="org.apache.log4j.RollingFileAppender">
            <param name="File" value="d:/log/warn.log" />
            <param name="MaxFileSize" value="5120KB" />
            <param name="MaxBackupIndex" value="10" />
            <layout class="org.apache.log4j.PatternLayout">
                    <param name="ConversionPattern"
                            value="%d{yyy-MM-dd HH:mm:ss} [%-
5p] %c {%F:%L} - %m%n" />
            </layout>
            <filter
class="org.apache.log4j.varia.LevelRangeFilter">
                    <param name="LevelMin" value="WARN" />
                    <param name="LevelMax" value="WARN" />
            </filter>
    </appender>

    <!--将级别为ERROR的信息输出到控制台-->
    <appender name="LOG.ERROR"
            class="org.apache.log4j.RollingFileAppender">
            <param name="File" value="d:/log/error.log" />
            <param name="MaxFileSize" value="5120KB" />
            <param name="MaxBackupIndex" value="10" />
            <layout class="org.apache.log4j.PatternLayout">
                    <param name="ConversionPattern"
                            value="%d{yyy-MM-dd HH:mm:ss} [%-
5p] %c {%F:%L} - %m%n" />
            </layout>
            <filter
class="org.apache.log4j.varia.LevelRangeFilter">
                    <param name="LevelMin" value="ERROR" />
                    <param name="LevelMax" value="ERROR" />
            </filter>
    </appender>

    <!--将级别为FATAL的信息输出到控制台-->
    <appender name="LOG.FATAL"
            class="org.apache.log4j.RollingFileAppender">
            <param name="File" value="d:/log/fatal.log" />
            <param name="MaxFileSize" value="5120KB" />
            <param name="MaxBackupIndex" value="10" />
            <layout class="org.apache.log4j.PatternLayout">
                    <param name="ConversionPattern"
                            value="%d{yyy-MM-dd HH:mm:ss} [%-
5p] %c {%F:%L} - %m%n" />
            </layout>
            <filter
class="org.apache.log4j.varia.LevelRangeFilter">
                    <param name="LevelMin" value="FATAL" />
                    <param name="LevelMax" value="FATAL" />
            </filter>
    </appender>

    <!--
            <appender name="InitAction"
class="org.apache.log4j.DailyRollingFileAppender">
            <param name="File" value="d:/dbcon.log"/>
            <param name="MaxFileSize" value="5120KB"/>
            <param name="MaxFileSize" value="10" />
```

```
            <param name="MaxBackupIndex" value="2" />
            <param name="DatePattern" value="'.'yyyy-MM-
dd'.'log"/>
            <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%d{yyy-MM-dd
HH:mm:ss} - %m%n"/>
            <param name="ConversionPattern" value="%d{DATE} [%-
5p] %c {%F:%L} - %m%n"/>
            </layout>
            </appender>
    -->

    <!--设置org.logicalcobwebs包的输出级别为INFO-->
    <!--
            <category name="org.logicalcobwebs">
            <priority value="INFO" />
            <appender-ref ref="LOG.Console" />
            </category>
    -->
    <root>
            <priority value="DEBUG" />
            <!--
                    <appender-ref ref="LOG.Console" />
            -->
            <appender-ref ref="LOG.DEBUG" />
            <appender-ref ref="LOG.INFO" />
            <appender-ref ref="LOG.WARN" />
            <appender-ref ref="LOG.ERROR" />
            <appender-ref ref="LOG.FATAL" />
    </root>
</log4j:configuration>
```

1.1.8. 设置单个包的输出级别

- Xml配置文件为在根节点前添加如下：

```
<!--设置org.logicalcobwebs包的输出级别为INFO-->
    <category name="org.logicalcobwebs">
            <priority value="INFO" />
            <appender-ref ref="LOG.Console" />
    </category>
```

- Properties文件的配置

```
log4j.logger.com.ibatis=DEBUG
log4j.logger.com.ibatis.common.jdbc.SimpleDataSource=DEBUG
log4j.logger.com.ibatis.common.jdbc.ScriptRunner=DEBUG
log4j.logger.com.ibatis.sqlmap.engine.impl.SqlMapClientDelegate
=DEBUG
log4j.logger.java.sql.Connection=DEBUG
log4j.logger.java.sql.Statement=DEBUG
log4j.logger.java.sql.PreparedStatement=DEBUG
log4j.logger.java.sql.ResultSet=DEBUG
```

1.1.9. 乱码问题

在linux平台下面log4j的输出会出现乱码，在每个块的下面设置如下：

```
log4j.appender.ALLINFO.Threshold=DEBUG
log4j.appender.ALLINFO.encoding=GBK
```

```
log4j.appender.ALLINFO=org.apache.log4j.DailyRollingFileAppende
r
```

1.1.10. 日志输出目录

在weblogic中设置log4j的日志输出目录可以为：

```
log4j.appender.ALLINFO.File=../BSM_Domain/servers/AdminServer/l
ogs/BSM/ALLINFO
```

或者

```
log4j.appender.ALLINFO.File=./servers/AdminServer/logs/BSM/ALLI
NFO
```

如果直接设置为

```
log4j.appender.ALLINFO.File = ALLINFO
```

则是在domain的根目录下面，即在BSM_Domain下面

加上上面的后就是在目录BSM_Domain/servers/AdminServer/logs/BSM/ALLINFO

下面。