Nutzung von Log4J und Commons Logging

Ausgaben steuern mit Commons Logging und Log4J

by Peter Roßbach

NOTICE:

In der Centaurus Platform gibt es diverse Ausgabemechanismen zu steuern. Die meisten heutigen Anwendungen und Frameworks steuern Ihre Ausgabe mit Commons Logging oder Log4J aus. Auch der Tomcat macht von diesem API regen Gebrauch. Interessant ist das es eine Trennung von Informationsbereitstellung und Ausgabe gibt.

1. Was ist Commons Logging und Log4J?

Das Commons Logging Projekt ist ein Logging API das sich darauf konzentiert die Information in der Anwendung zu formulieren und von dem Ausgabe-Framework zu abstrahieren. Als einer der Ausgabe-Frameworks wird Log4j unterstützt.

2. Installation

Folgende Dateien und Verzeichnisse müssen besorgt, konfiguiert und installiert werden:

```
${centaurus.base}
|--libs
        |--common
                |--lib
                        |--log4j-1.2.8.jar
                        |--commons-logging-1.0.4.jar
|--conf
        |--log4j.xml
```

Als letzten Schritt erfolgt nun die Konfiguration in der Datei *conf/server.xml* des Listener:

```
<Listener
    className="de.centaurus.platform.catalina.ext.logger.Log4JWatcherLifecycleListener"
    enabled="true"
    configuration="conf/log4j.xml"
    checkInterval="60"
    >
 </Listener>
```

3. Der Log4JWatcherLifecycleListener

Die Centaurus-Platform stellt für die Ausgabesteuerung mit Log4J eine eigenen Konfigruationswächter zur Verfügung: den *Log4JWatcherLifecycleListener*

Dieser Listener konfiguiert den Default Logger als JMX MBean und prüft die Konifgruation in einem einstellbaren Intervall auf Neuigkeiten.

3.1. Parameter

className	Der Klassennamen des Log4jWatchers. Hier sollten Sie den Wert **de.centaurus.platform.catalina.ext.logger.Log4JWatcherLifecy** benutzen.
enabled	Mit dem Wert *true* wird der Watcher aktiv. (Default *true*)
configuration	Hier kann der Pfad zur log4j Konfigruation angeben werden. Es kann entweder eine XML oder Properties Datei gewählt werden. (Default: **${catalina.base}/conf/log4j.xml**) benutzen.
checkInterval	Mit diesem Wert wird das Prüfintervall in Sekunden angeben. Der Wert *-1* unterdrückt die Beobachtung.(Default *60*)

3.2. Beispiel Log4J-Konfiguration

```
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/" debug="true">
<!-- ================Appenders for CSF=================-->
<appender name="CATALINA" class="org.apache.log4j.RollingFileAppender">
     <param name="Threshold" value="DEBUG"/>
     <param name="File" value="logs/catalina.log"/>
     <param name="Append" value="true"/>
     <param name="MaxFileSize" value="5000KB"/>
     <param name="MaxBackupIndex" value="2"/>
     <layout class="org.apache.log4j.PatternLayout">
```

```
          <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>
     </layout>
</appender>

<!-- ============================== -->
<!-- Append messages to the console -->
<!-- ==============================-->

<appender name="CONSOLE" class="org.apache.log4j.ConsoleAppender">
     <param name="Target" value="System.out"/>
     <param name="Threshold" value="INFO"/>
     <layout class="org.apache.log4j.PatternLayout">
          <!--The default pattern: Date Priority [Category] Message\n-->
          <param name="ConversionPattern" value="%d{ABSOLUTE} %-5p [%c{1}] %m%n"/>
     </layout>
</appender>

<!-- ======================== -->
<!-- Limit categories server  -->
<!-- ======================== -->
<!--
RULES for logging DEBUG < INFO < WARN < ERROR < FATAL.
-->
<!-- catalia -->
<category
     name="org.apache.catalina"
     additivity="true">
         <priority value="info" />
     <appender-ref ref="CATALINA"/>
</category>

<category
     name="org.apache.tomcat"
     additivity="true">
         <priority value="info" />
     <appender-ref ref="CATALINA"/>
</category>

<!-- ======================== -->
<!-- Limit categories apps    -->
<!-- ======================== -->

<category
     name="org.objektpark.tomcat.hello"
     additivity="true">
         <priority value="info" />
     <appender-ref ref="CONSOLE"/>
</category>

<!-- Setup the Root console  -->
<root>
<appender-ref ref="CONSOLE"/>
</root>
</log4j:configuration>
```

Im Tomcat 5.next werden die Logger Element des Tomcats entfernt und die Umstellung auf Commons Logging komplett vollzogen. Auf folgende Category wird dann die Ausgabe gesteuert:

```
<category
      name="org.apache.catalina.core.ContainerBase.[Catalina]"
      additivity="false">
          <priority value="info" />
      <appender-ref ref="catalina" />
</category>
<category
      name="org.apache.catalina.core.ContainerBase.[Catalina].[localhost]"
      additivity="false">
          <priority value="info" />
      <appender-ref ref="localhost" />
</category>
<category
      name="org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager]"
      additivity="false">
          <priority value="info" />
      <appender-ref ref="localhost.manager" />
</category>
```

4. Der CommonsLoggingLogger

Die Centaurus-Platform stellt für die Ausgabe des Tomcats noch einen speziellen Logger zur verfügung. Hiermit können sie dann die gesamte Ausgabe des Server auf Commons Logging API umstellen.
Folgende Parameter werden unterstützt

className	Der Klassennamen des Tomcat Loggers. Hier sollten Sie den Wert **de.centaurus.platform.catalina.ext.logger.CommonsLoggingL** benutzen.
loggerName	Name des Logger in der Log4J Konfiguration
verbosityLevel	Ausgabe Level dieses Logger mit Namen. (*FATAL* < *ERROR* < *WARNING* < *INFORMATION* < *DEBUG* <)

Beispiel für einen CommonsLoggingLogger

```
<Logger
     className="de.centaurus.platform.catalina.ext.logger.CommonsLoggingLogger"
     loggerName="org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manage
     verbosityLevel="DEBUG"
```

```
/>
```

Note:

Leider wird die Aktualieserung aus der Datei noch nicht in die MBeans korrekt übernommen.

Note:

In einem späteren Release sollen alle Element der Log4J-Konfiguration als MBeans aktuell bereit stehen. Auch ein Speicherung des aktuellen Log4J Konfiguration wäre hilfreich.

Note:

Wenn Web-Anwendnungen ihre eigenen Log4J Klassen und Konfiguration mitbringen ist eine Aktualliserung bisher nicht möglich.