

# Configuring LOG4J for MoSKito

After reading this section, you will know...

... how to make MoSKito work with LOG4J Logging Framework.

In this section:

- [Create a logger](#)
- [Configure Log4J](#)
  - [Define Appenders](#)
  - [Define Logger](#)
- [What else you might want to know](#)

MoSKito ships with embedded support for major Logging frameworks, in particular **log4j**, **java.util.logging** and, for lazy people, **System.outs**. In order to attach a logger to a [StatsProducer](#), follow the steps below:



Please note, MoSKito 2.3.0 comes with SLF4J instead of LOG4J.

## Create a logger

Moskito comes with two built-in 'loggers', the **IntervalStatsLogger** and the **DefaultStatsLogger**.

The **IntervalStatsLogger** can be attached to an Interval, the **DefaultStatsLogger** attaches itself to **StatsProducer**, that can't be reset. Both are connecting to the internal **IntervalUpdateService** and monitor interval updates. As soon as the **IntervalUpdate** is fired, they read the current state of the Producer they are attached to and send it to the **LogOutput**.

Below is an example of how to configure and attach a **log4j** logger to any producer:

```
import org.apache.log4j.Logger;
import net.java.dev.moskito.core.logging.IntervalStatsLogger;
import net.java.dev.moskito.core.logging.DefaultStatsLogger;
import net.java.dev.moskito.core.logging.Log4JOutput;
import net.java.dev.moskito.core.stats.DefaultIntervals;

....
IStatsProducer producer = ...;
// add moskito logger
new DefaultStatsLogger(producer, new Log4JOutput(Logger.getLogger("MoskitoDefault")));
new IntervalStatsLogger(producer, DefaultIntervals.FIVE_MINUTES, new Log4JOutput(Logger.getLogger("Moskito5m")));
new IntervalStatsLogger(producer, DefaultIntervals.FIFTEEN_MINUTES, new Log4JOutput(Logger.getLogger("Moskito15m")));
new IntervalStatsLogger(producer, DefaultIntervals.ONE_HOUR, new Log4JOutput(Logger.getLogger("Moskito1h")));
new IntervalStatsLogger(producer, DefaultIntervals.ONE_DAY, new Log4JOutput(Logger.getLogger("Moskito1d")));
//end moskito logger
```

## Configure Log4J

Attaching a **log4j** logger to a MoSKito producer is only half of the work, you still need to tell **log4j** where to log it.

Below is an example that creates local files for each interval and logs into them:

## Define Appenders

## log4j.xml

```
<!-- MOSKITO APPENDER DECLARATION START -->

<appender name="MoskitoDefaultFileAppender" class="org.apache.log4j.RollingFileAppender">
  <param name="File" value="logs/monitoring_msk.log" />
  <param name="MaxFileSize" value="100MB" />
  <param name="MaxBackupIndex" value="5" />
  <param name="Threshold" value="INFO" />
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern"
      value="%r %d{ISO8601} %m%n"/>
  </layout>
</appender>

<appender name="Moskito5mFileAppender" class="org.apache.log4j.RollingFileAppender">
  <param name="File" value="logs/monitoring_msk_5m.log" />
  <param name="MaxFileSize" value="100MB" />
  <param name="MaxBackupIndex" value="5" />
  <param name="Threshold" value="INFO" />
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern"
      value="%r %d{ISO8601} %m%n"/>
  </layout>
</appender>

<appender name="Moskito15mFileAppender" class="org.apache.log4j.RollingFileAppender">
  <param name="File" value="logs/monitoring_msk_15m.log" />
  <param name="MaxFileSize" value="100MB" />
  <param name="MaxBackupIndex" value="5" />
  <param name="Threshold" value="DEBUG" />
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern"
      value="%r %d{ISO8601} %m%n"/>
  </layout>
</appender>

<appender name="Moskito1hFileAppender" class="org.apache.log4j.RollingFileAppender">
  <param name="File" value="logs/monitoring_msk_1h.log" />
  <param name="MaxFileSize" value="100MB" />
  <param name="MaxBackupIndex" value="5" />
  <param name="Threshold" value="INFO" />
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern"
      value="%r %d{ISO8601} %m%n"/>
  </layout>
</appender>

<appender name="MoskitoldFileAppender" class="org.apache.log4j.RollingFileAppender">
  <param name="File" value="logs/monitoring_msk_ld.log" />
  <param name="MaxFileSize" value="100MB" />
  <param name="MaxBackupIndex" value="5" />
  <param name="Threshold" value="INFO" />
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern"
      value="%r %d{ISO8601} %m%n"/>
  </layout>
</appender>

<!-- /Moskito Appender DECLARATION END -->
```

## Define Logger

## log4j.xml

```
<!-- Moskito LOGGER DECLARATION START -->

<logger name="MoskitoDefault" additivity="false">
  <level value="INFO"/>
  <appender-ref ref="MoskitoDefaultFileAppender"/>
</logger>

<logger name="Moskito5m" additivity="false">
  <level value="INFO"/>
  <appender-ref ref="Moskito5mFileAppender"/>
</logger>

<logger name="Moskito15m" additivity="false">
  <level value="INFO"/>
  <appender-ref ref="Moskito15mFileAppender"/>
</logger>

<logger name="Moskito1h" additivity="false">
  <level value="INFO"/>
  <appender-ref ref="Moskito1hFileAppender"/>
</logger>

<logger name="Moskitold" additivity="false">
  <level value="INFO"/>
  <appender-ref ref="MoskitoldFileAppender"/>
</logger>

<!-- /Moskito LOGGER DECLARATION END -->
```

## What else you might want to know

- The **DefaultStatsLogger** logs throughout 60 seconds. The **IntervalStatsLogger** logs at the end of the interval.
- The LogLevel is *info*.