

How-to enable/configure logging?

SNMP4J provides its own log framework adapter that allows to plugin virtually any logging framework, including Java logging and Log4J.

Enable Simple Logging

In the main class execute the following code before any SNMP4J code is executed statically or at runtime:

```
static {
    LogFactory.setLogFactory(new ConsoleLogFactory());
    ConsoleLogAdapter.setDebugEnabled(true);
}
```

Enable Log4J Logging

In the main class put:

```
static {
    LogFactory.setLogFactory(new Log4jLogFactory());
    org.apache.log4j.BasicConfigurator.configure();
    LogFactory.getLogFactory().getRootLogger().setLogLevel(LogLevel.ALL);
}
```

Enable Java Logging

To enable Java logging two steps are necessary:

Code

```
static {
    LogFactory.setLogFactory(new JavaLogFactory());
    // Optionally set log level on root logger:
    LogFactory.getLogFactory().getRootLogger().setLogLevel(LogLevel.ALL);
}
```

Runtime Configuration

For SNMP4J-CLT 2.1 you need to configure the log handlers and format at runtime. In SNMP4J and any other SNMP4J-* API you can do this with code of course.

Add the parameter `-Djava.util.logging.config.file=<file>` to the Java command line:

```
java -Djava.util.logging.config.file=java_logging.properties ...
```

java_logging.properties Sample Configuration

```
handlers = java.util.logging.ConsoleHandler
java.util.logging.ConsoleHandler.level = ALL
java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter
java.util.logging.SimpleFormatter.format=%1$tY-%1$tm-%1$td %1$tH:%1$tM:%1$tS %4$-6s %5$s%6$s%n
```