```
import org.apache.log4j.Logger;
public class MonitorInfractions {
private static Logger log=
Logger.getLogger(ClientLogToStreams.class.getPackage().getName());
       log.info("Message from program testing');.
                          log4i - config
log4j.logger.com.ibm.streamsx.log4j.example.client=DEBUG,SOCKET
log4j.appender
              import org.apache.log4j.Logger;
pg4j.appende
              public class ProcessCellularRecords {
g4j.appende
              private static Logger log=
              Logger.getLogger(ClientLogToStreams.class.getPackage().getName());
                     log.info("Message from program Info");....
                                        log4j - config
    Network
             log4j.logger.com.ibm.streamsx.log4j.example.client=DEBUG,SOCKET
              log4j.appender.SOCKET=org.apache.log4j.net.SocketAppender
              log4j.appender.SOCKET.remoteHost=localhost
              log4i.appender.SOCKET.port=24400
                                                   LogSink
                  Log4j
                      Operator: com.ibm.streamsx.log4j::ReceiveLog4j Log4j
                      Source:
                            stream<rstring message> Log4j = ReceiveLog4j()
                                param
                                   port:24400;
                                   log4jConfig:"streams.config";
                             }
                        log4j_rootLogger=DEBUG,Xstream
                        # set the appender named X to b a xonsole appender
                        log4j.appender.Xstream=org.apache.log4j.ConsoleAppender
                        # set layout
                        log4j.appender.Xstream=tk.streams.log4j.StreamsAppender
                        log4j.appender.Xstream.layout=org.apache.log4j.PatternLayout
                        og4j.appender.Xstream.layout.conversionPattern=>>>>>>>%m%n
```