

# KIP-521: Enable redirection of Connect's log4j messages to a file by default

- [Status](#)
- [Motivation](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Test Plan](#)
- [Rejected Alternatives](#)

## Status

**Current state:** *Adopted.*

**Discussion thread:** [here](#)

**Vote thread:** [here](#)

**JIRA:** [KAFKA-5609](#)

**Released:** *N/A*

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

## Motivation

Kafka Connect workers, similarly to Kafka brokers, are long running processes that accept configuration properties from the command line and print logs using a logging framework such as log4j or similar. However, the default setup of the logging framework for Connect redirects the logs of the service by default only to the standard output. Although someone can configure the logging framework to redirect the logs to a file (as well as to any custom logger), a configuration that captures out-of-the-box the most common and useful use case and also provides consistent experience across Kafka components is preferable.

## Proposed Changes

This proposal suggests to change Connect's log4j properties file to instruct the logging framework to append log lines into a file as well as the console (standard output). An additional logger will be defined for this purpose, along with settings that will allow for the periodic rotation of these log files. The suggested new log4j properties are shown below.

### New connect-log4j.properties to allow rotating export to a file

```
log4j.rootLogger=INFO, stdout, connectAppender

log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.connectAppender=org.apache.log4j.DailyRollingFileAppender
log4j.appender.connectAppender.DatePattern='.'yyyy-MM-dd-HH
log4j.appender.connectAppender.File=${kafka.logs.dir}/connect.log
log4j.appender.connectAppender.layout=org.apache.log4j.PatternLayout

# The `%X{connector.context}` parameter in the layout includes connector-specific and task-specific information
# in the log message, where appropriate. This makes it easier to identify those log messages that apply to a
# specific connector. Simply add this parameter to the log layout configuration below to include the contextual
# information.
#
connect.log.pattern=[%d] %p %m (%c:%L)%n
#connect.log.pattern=[%d] %p %X{connector.context}%m (%c:%L)%n

log4j.appender.stdout.layout.ConversionPattern=${connect.log.pattern}
log4j.appender.connectAppender.layout.ConversionPattern=${connect.log.pattern}

log4j.logger.org.apache.zookeeper=ERROR
log4j.logger.org.reflections=ERROR
```

## Compatibility, Deprecation, and Migration Plan

The proposed change is backwards compatible because:

- Directing logs to the console (standard output) is preserved.
- Using the environment variable `\${kafka.logs.dir}` is a compatible addition, since this variable is already used within `kafka-run-class.sh` which is the base script that is used by Kafka Connect's startup scripts. Using this variable is compatible with setups that customized log4j configuration by the environment variable `LOG_DIR` previously

## Test Plan

Setting up regular tests for this change does not seem necessary, especially given that the approach follows what's already common practice in Kafka brokers. The changes will be tested manually and soak tested in long running Connect clusters.

## Rejected Alternatives

The suggested changes are fairly simple. No other potential alternatives have been considered.