**LOG4J**  🡪MIGRATION🡪 **LOG4J2**

**Reasons to upgrade from Log4j 1.x to Log4j 2**

* Community support for Log4j 2.
* [Async Loggers](http://logging.apache.org/log4j/2.x/manual/async.html) - performance [similar to logging switched off](http://blogs.mulesoft.com/dev/mule-dev/mule-3-6-asynchronous-logging/)
* [Custom log levels](http://logging.apache.org/log4j/2.x/manual/customloglevels.html)
* Automatically reload its configuration upon modification without losing log events while reconfiguring.
* Java 8-style [lambda support](https://logging.apache.org/log4j/2.x/manual/api.html#LambdaSupport) for lazy logging
* Log4j 2 is [garbage-free](https://logging.apache.org/log4j/2.x/manual/garbagefree.html) (or at least low-garbage) since version 2.6
* [Filtering](https://logging.apache.org/log4j/2.x/manual/filters.html): Based on context data, markers, regular expressions, and other components in the Log event. Filters can be associated with Loggers.
* [Plugin Architecture](https://logging.apache.org/log4j/2.x/manual/plugins.html) - easy to extend by building custom components
* Supported APIs: SLF4J, Commons Logging, Log4j-1.x and java.util.logging
* Log4j 2 API separate from the Log4j 2 implementation. API supports more than just logging Strings: CharSequences, Objects and custom [Messages](https://logging.apache.org/log4j/2.x/manual/messages.html). Messages allow support for interesting and complex constructs to be passed through the logging system and be efficiently manipulated. Users are free to create their own Message types and write custom Layouts, Filters and Lookups to manipulate them.
* Concurrency improvements: log4j2 uses java.util.concurrent libraries to perform locking at the lowest level possible. Log4j-1.x has some deadlock issues.
* Configuration via XML, JSON, YAML, properties configuration files or programmatically.

**Be aware**

* log4j2.xml and log4j2.properties formats are different from the Log4j 1.2 configuration syntax
* Log4j 2 is not [fully](http://logging.apache.org/log4j/2.x/manual/migration.html) compatible with Log4j 1.x: The Log4j 1.2 API is supported by the log4j-1.2-api adapter but customizations that rely on Log4j 1.2 internals may not work.
* Java 6 required for version 2.0 to 2.3. Java 7 is required for Log4j 2.4 and later.

**Tips when upgrading**

(Common issues people are having when getting started with log4j2):

* You need ([at least](http://logging.apache.org/log4j/2.x/faq.html#which_jars)) both log4j-api-2.6.2.jar and log4j-core-2.6.2.jar in your classpath
* Log4j2 looks for a log4j**2**.xml config file, not a log4j.xml config file
* Config file [location](http://logging.apache.org/log4j/2.x/faq.html#config_location): either put it in the classpath or specify its path with the log4j.configurationFile system property
* To [debug the configuration](http://logging.apache.org/log4j/2.x/faq.html#troubleshooting), use <Configuration status="trace"> in the beginning of your config file

**Log4j 2 Compatibility with Log4j 1 (If we try using both):**

* <https://logging.apache.org/log4j/2.x/manual/compatibility.html>

**Ways to migrate:**

* **Using the Log4j 1.x bridge**
* Replace the log4j 1.x jar file with Log4j 2's log4j-1.2-api.jar.
* Requirements:
* They must not access methods and classes internal to the Log4j 1.x implementation such as Appenders, LoggerRepository or Category's callAppenders method.
* They must not programmatically configure Log4j.
* They must not configure by calling the classes DOMConfigurator or PropertyConfigurator.
* **Converting to the Log4j 2 API**
* Steps:
  + Change org.apache.log4j to org.apache.logging.log4j , accordingly the corresponding api calls.
  + Some Getter methods have been shifted to LogManager class.
  + Calls to org.apache.log4j.Logger.getLogger that accepts  LoggerFactory must remove the  org.apache.log4j.spi.LoggerFactory and use one of Log4j 2's other extension mechanisms.
  + Remove calls to org.apache.log4j.LogManager.shutdown(), now Log4j Core automatically adds a JVM shutdown hook on start up to perform any Core clean ups.
  + Calls to org.apache.log4j.Logger.setLevel() or similar methods are not supported in the API. Applications should remove these. Equivalent functionality is provided in the Log4j 2 implementation classes but may leave the application susceptible to changes in Log4j 2 internals.
* Sample properties file-
  + from log4j:

log4j.rootLogger=INFO, R

log4j.appender.R = org.apache.log4j.DailyRollingFileAppender

log4j.appender.R.File = /trace.log

log4j.appender.R.Append = true

log4j.appender.R.DatePattern = '.'yyyy-MM-dd

log4j.appender.R.layout = org.apache.log4j.PatternLayout

log4j.appender.R.layout.ConversionPattern = %d{HH:mm:ss} %c{1} [%p] %m%n

* + to log4j2:

rootLogger.level = INFO

property.filename = trace.log

appenders = R, console

appender.console.type = Console

appender.console.name = STDOUT

appender.console.layout.type = PatternLayout

appender.console.layout.pattern = %d %5p [%t] (%F:%L) - %m%n

appender.R.type = RollingFile

appender.R.name = File

appender.R.fileName = ${filename}

appender.R.filePattern = ${filename}.%d{yyyy-MM-dd}

appender.R.layout.type = PatternLayout

appender.R.layout.pattern = %d{yyyy-MM-dd HH:mm:ss} %c{1} [%p] %m%n

appender.R.policies.type = Policies

appender.R.policies.time.type = TimeBasedTriggeringPolicy

appender.R.policies.time.interval = 1

rootLogger.appenderRefs = R, console

rootLogger.appenderRef.console.ref = STDOUT

rootLogger.appenderRef.R.ref = File

* For more, refer: <https://logging.apache.org/log4j/2.x/manual/migration.html> (Includes Sample Configurations for Log4J and corresponding config for Log4J2).
* Log4j2 supports .properties files but they have changed property syntax. Refer [manual here](https://logging.apache.org/log4j/2.0/manual/configuration.html#Properties) (<https://logging.apache.org/log4j/2.0/manual/configuration.html#Properties>), it covers how to create new configuration.
* Other Links that may be useful:
  + Convert from v1.2 properties to v1.2XML using this converter: <https://log4j-props2xml.appspot.com/>
  + <https://www.optimit.hr/en/blogs/log4j-to-log4j2-migration/>
  + <https://www.baeldung.com/java-logging-intro>
  + Log4J2 Hello World Tutorial: <https://mkyong.com/logging/apache-log4j-2-tutorials/> **[IMP]**
  + Official guide to LOG4J2 configuration: <https://logging.apache.org/log4j/2.x/manual/configuration.html>
  + JournalDev Log4J 2 guide <https://www.journaldev.com/7128/log4j2-example-tutorial-configuration-levels-appenders>
  + <https://stackoverflow.com/questions/30019585/log4j2-why-would-you-use-it-over-log4j>
  + <https://stackify.com/compare-java-logging-frameworks/>
  + <https://www.programmerall.com/article/92451405187/>
  + **Lambda Support:** <https://logging.apache.org/log4j/2.x/manual/api.html#LambdaSupport>
  + **Custom Log Levels:** <https://logging.apache.org/log4j/2.x/manual/customloglevels.html>
  + **Async Logging:** <https://logging.apache.org/log4j/2.x/manual/async.html>