Log4j 2 and Hibernate 4

As of this writing Hibernate 4.3.7.Final is the current version of Hibernate and it uses jboss-logging Version 3.1.3.GA for logging. jboss-logging 3.2.0.Beta2 will support log4j 2.x natively and no additional configuration will be required, but it hasn't been released, yet. Once it has been released, most information on this page will become obsolete.

As a sidenote: jboss-logging 3.2.0.Beta1 claims to support log4j 2.x natively but was based on beta releases of log4j 2.x - due to a backwards incompatible change in log4j 2.x after its beta jboss-logging 3.2.0.Beta1 doesn't work with the final release of log4j 2.x.

jboss-logging is yet another layer on top of popular logging frameworks. You can configure which logging framework it should delegate to using the system property org.jboss.logging.provider, if it is not set its discovery algorithm will try

- jBoss' own logging implementation
- log4j 2.x - starting with 3.2.0.Beta2
- log4j 1.x - only if some internal log4j classes can be found
- slf4j - only if logback is on the classpath as well
- java.util.logging

in that order. The possible values for the system property are jboss, log4j2, log4j, slf4j and jdk.

In order to use log4j2 with jboss-logging prior to 3.2.0.Beta2 you have two choices

- add the log4j 2.x slf4j binding and slf4j-api to your classpath as well and set the system property org.jboss.logging.provider to slf4j - setting the property explicitly bypasses the "logback is there" check
- override Hibernate's dependency to use jboss-logging 3.2.0.Beta1, add the log4j 1.x logging bridge to your classpath as well and set the system property org.jboss.logging.provider to log4j - this doesn't seem to work with any 3.1.x version of jboss-logging

Things get even more complex if you add a non-default connection pool to Hibernate. c3p0 prior to 0.9.5pre2 will only log to log4j 1.x so using the log4j 1.x logging bridge is your only option. Later versions of c3p0 added support for slf4j as well.