MyBatis的日志输出

# MyBatis的Logging日志输出

<http://www.mybatis.org/mybatis-3/logging.html>

# MyBatis的Logging介绍

MyBatis provides **logging information** through the use of an internal log factory. The internal log factory will delegate logging information to one of the following log implementations:

* SLF4J
* Apache Commons Logging
* Log4j 2
* Log4j
* JDK logging

**The logging solution chosen is based on runtime introspection by the internal MyBatis log factory.** The MyBatis log factory will use the first logging implementation it finds (implementations are searched in the above order). If MyBatis finds none of the above implementations, then logging will be disabled.

**MyBatis内部提供了Log工厂**。内部的**Log工厂**委托log实现来完成。MyBatis的log工厂会使用它所寻找到的第一个具体的实现，顺序就是**SLF4J、Apache Commons Logging、Log4j 2、Log4j、JDK logging**。如果没有找到任何一个，则就没有日志输出。

# Mybatis的日志配置步骤：MyBatis-Logging-Configuration

下面以使用log4j为例：

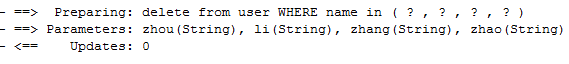
## Step1 添加log4j 的jar包

对于web工程，要把jar包放到WEB-INF/lib目录下，并添加到classpath下。

## Step2：Configure Log4J

主要编写**log4j.properties**文件，该文件不需要手动加载，只需要将该文件放置到**src目录或项目根目录**下即可。

与普通的日志区别就是添加**log4j.logger.xxx=TRACE 即可，声明MyBatis的日志输出作用范围，其他的没有任何区别。如：**



配置文件示例：

# Global logging configuration

log4j.rootLogger=**ERROR, stdout**

# MyBatis logging configuration...

######需要添加的地方#######

**log4j.logger.com.unisk.mapper=TRACE #该包下**

log4j.logger.**com.unisk.mapper.UserMapper**=TRACE#该mapper下

log4j.logger.com.unisk.mapper.UserMapper.insertUser=TRACE#仅该操作

######需要添加的地方#######

# Console output...

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

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

log4j.appender.stdout.layout.ConversionPattern=%5p [%t] - %m%n

# 日志输出的范围控制

**log4j.logger.xxx=TRACE**

**xxx决定了控制的范围，可以配置多个。**

T**o see MyBatis logging statements** you may enable logging on a package, a mapper fully qualified class name, a namespace or a fully qualified statement name.

可以是**输出一个包、mapper、名称空间、一个具体的操作（方法）**的日志信息。

### 包范围: log4j.logger.com.unisk.mapper=TRACE

在Mapper代理开发过程中，由于**namespace**一般取**包的名字**，所以就和包一样了。

### 类范围（Mapper接口）log4j.logger.com.unisk.mapper.UserMapper=TRACE

### 某个方法：（具体的某个操作）

log4j.logger.com.unisk.mapper.UserMapper.insertUser=TRACE

# 在全局配置文件中配置：logImpl

Many environments ship Commons Logging as a part of the application server classpath (good examples include Tomcat and WebSphere). It is important to know that in such environments, **MyBatis will use Commons Logging as the logging implementation.** In an environment like WebSphere this will mean that your Log4J configuration will be ignored **because WebSphere supplies its own proprietary implementation of Commons Logging**. **This can be very frustrating because it will appear that MyBatis is ignoring your Log4J configuration** (in fact, MyBatis is ignoring your Log4J configuration because MyBatis will use Commons Logging in such environments). If your application is running in an environment where Commons Logging is included in the classpath but you would rather use one of the other logging implementations you can select a different logging implementation **by adding a setting in mybatis-config.xml file as follows.**

