**Logger Hibernate**

**1. Using Log4j.xml**

**2. Using Log4j.properties**

**Jars :- log4j.jar and slf4j.jar**

**Debugger Level**

We have used DEBUG with both the appenders. All the possible options are:

* TRACE
* DEBUG
* INFO
* WARN
* ERROR
* FATAL
* ALL

## Appenders in Logger: Apache log4j provides Appender objects which are primarily responsible for printing logging messages to different destinations such as consoles, files, sockets, NT event logs, etc.

## Each Appender object has different properties associated with it.

|  |  |
| --- | --- |
| **Property** | **Description** |
| Layout | Appender uses the Layout objects and the conversion pattern associated with them to format the logging information. |
| Target | The target may be a console, a file, or another item depending on the appender. |
| Level | The level is required to control the filtration of the log messages. |
| Threshold | Appender can have a threshold level associated with it independent of the logger level. The Appender ignores any logging messages that have a level lower than the threshold level. |
| Filter | The Filter objects can analyze logging information beyond level matching and decide whether logging requests should be handled by a particular Appender or ignored. |

We can add an Appender object to a Logger by including the following setting in the configuration file with the following method:

log4j.logger.[logger-name]=level, appender1,appender..n

You can write same configuration in XML format as follows:

<logger name="com.apress.logging.log4j" additivity="false">

<appender-ref ref="appender1"/>

<appender-ref ref="appender2"/>

</logger>

We have used only one appender *FileAppender* in our example above. All the possible appender options are:

* AppenderSkeleton
* AsyncAppender
* ConsoleAppender
* DailyRollingFileAppender
* ExternallyRolledFileAppender
* FileAppender
* JDBCAppender
* JMSAppender
* LF5Appender
* NTEventLogAppender
* NullAppender
* RollingFileAppender
* SMTPAppender
* SocketAppender
* SocketHubAppender
* SyslogAppender
* TelnetAppender
* WriterAppender

We would cover FileAppender in [Logging in Files](https://www.tutorialspoint.com/log4j/log4j_logging_files.htm) and JDBC Appender would be covered in [Logging in Database](https://www.tutorialspoint.com/log4j/log4j_logging_database.htm)

**Layout**

We have used PatternLayout with our appender. All the possible options are:

* DateLayout
* HTMLLayout
* PatternLayout
* SimpleLayout
* XMLLayout

Using HTMLLayout and XMLLayout, you can generate log in HTML and in XML format as well.