**Java Logging: Configuration**

The Java Logging API can be configured in two ways:

1. Via a configuration class.
2. Via a configuration file.

The initialization of the configuration is taken care of by the java.util.logging.LogManager class.

**Configuration Class**

You can use a Java class to configure the Java Logging API. You do so by specifying the name of the class in the JVM parameter java.util.logging.config.class. It is the constructor of that class that should load the configuration and apply it to the Logger's in the hierarchy.

**Configuration File**

If no configuration class is specified, you can instead specify a configuration file (but no configuration class can be specified then!).

The Java Logging API has a default logging configuration file located at "lib/logging.properties", inside the JRE directory. If you edit this file, you edit the default logging settings for the entire JRE, for every program executed. This is most often not what you want to do, though.

Instead, you can set a separate configuration file for your application. You do so by setting the JVM propertyjava.util.logging.config.file to point to this file.

The configuration file is a standard property file as you know them from Java. Inside this property file you can set properties that configure the various Logger's and Handler's used in your application.

Here is a list of properties you can set in the configuration file. You should double check the JavaDoc over time to see if any of this changes (e.g. in a later version of Java than Java 6).

|  |  |
| --- | --- |
| Property | Description |
| handlers | A white space or comma separated list of handler class names to be added to the root Logger |
| config | A white space or comma separated list of class names which will be instantiated when the LogManager is initialized. The constructors of these classes can execute arbitrary configuration code. |
| "logger".handlers | Sets the handler classes to use for a givenLogger in the hierarchy. Replace the "logger" with a specific name of a Logger in your app (e.g. com.jenkov.web). |
| "logger".useParentHandlers | Tells a given Logger whether it should log to its parents or not (true or false). |
| "logger".level | Tells a given Logger what minimum log level it should log. |
|  |  |
| java.util.logging.FileHandler.level | Sets the default log level for all FileHandler's. |
| java.util.logging.FileHandler.filter | A class name of the Filter to use on allFileHandler's. |
| java.util.logging.FileHandler.formatter | A class name of the Formatter to use on allFileHandler's. |
| java.util.logging.FileHandler.encoding | The encoding to use by all FileHandler's (e.g. UTF-8, UTF-16 etc.). |
| java.util.logging.FileHandler.limit | The approximate amount of bytes to write to a log file, before rotating to a new file. |
| java.util.logging.FileHandler.count | The number of log files to use in the log file rotation. |
| java.util.logging.FileHandler.append | Sets whether or not the FileHandler's should append to an existing file or not (true or false), if an existing log file is found. |
| java.util.logging.FileHandler.pattern | The log file name pattern. |
|  |  |
| java.util.logging.ConsoleHandler.level | Sets the default log level of allConsoleHandler's. |
| java.util.logging.ConsoleHandler.filter | Sets the Filter to use by allConsoleHandler's |
| java.util.logging.ConsoleHandler.formatter | Sets the Formatter to use by allConsoleHandler's. |
| java.util.logging.ConsoleHandler.encoding | Sets the encoding to use by allConsoleHandler's. |
|  |  |
| java.util.logging.StreamHandler.level | Sets the default log level of all StreamHandler's. |
| java.util.logging.StreamHandler.filter | Sets the Filter to use by all StreamHandler's |
| java.util.logging.StreamHandler.formatter | Sets the Formatter to use by allStreamHandler's. |
| java.util.logging.StreamHandler.encoding | Sets the encoding to use by allStreamHandler's. |
|  |  |
| java.util.logging.SocketHandler.level | Sets the default log level of all SocketHandler's. |
| java.util.logging.SocketHandler.filter | Sets the Filter to use by allSocketHandler's. |
| java.util.logging.SocketHandler.formatter | Sets the Formatter to use by allSocketHandler's. |
| java.util.logging.SocketHandler.encoding | Sets the encoding to use by allSocketHandler's. |
| java.util.logging.SocketHandler.host | Sets the host name of the host to send the log messages to (e.g. jenkov.com). |
| java.util.logging.SocketHandler.port | Sets the port number of of the host to send the log message to (e.g. 9999). |
|  |  |
| java.util.logging.MemoryHandler.level | Sets the default log level of all MemoryHandler's. |
| java.util.logging.MemoryHandler.filter | Sets the Filter to use by allMemoryHandler's. |
| java.util.logging.MemoryHandler.size | The size of the internal LogRecord buffer. |
| java.util.logging.MemoryHandler.push | The push level of messages causing the buffer to be pushed to the target Handler. Defaults to SEVERE. |
| java.util.logging.MemoryHandler.target | The class name of the target Handler. |

Here is an example configuration file. Not all properties are set to a value. You should do that in your own configuration file, or omit the configuration parameter if it has no value.

Note: Configuration properties are applied in the sequence they are listed in the config file. That means, that you should configure parent Logger's before child Logger's. Otherwise the configuration of the parent Logger will override that of the child Logger.

handlers = java.util.logging.FileHandler

config =

"logger".handlers =

"logger".useParentHandlers =

"logger".level =

java.util.logging.FileHandler.level = WARNING

java.util.logging.FileHandler.filter =

java.util.logging.FileHandler.formatter =

java.util.logging.FileHandler.encoding =

java.util.logging.FileHandler.limit =

java.util.logging.FileHandler.count =

java.util.logging.FileHandler.append = false

java.util.logging.FileHandler.pattern = log.%u.%g.txt

java.util.logging.ConsoleHandler.level = WARNING

java.util.logging.ConsoleHandler.filter =

java.util.logging.ConsoleHandler.formatter =

java.util.logging.ConsoleHandler.encoding =

java.util.logging.StreamHandler.level = WARNING

java.util.logging.StreamHandler.filter =

java.util.logging.StreamHandler.formatter =

java.util.logging.StreamHandler.encoding =

java.util.logging.SocketHandler.level = WARNING

java.util.logging.SocketHandler.filter =

java.util.logging.SocketHandler.formatter =

java.util.logging.SocketHandler.encoding =

java.util.logging.SocketHandler.host =

java.util.logging.SocketHandler.port =

java.util.logging.MemoryHandler.level = WARNING

java.util.logging.MemoryHandler.filter =

java.util.logging.MemoryHandler.size =

java.util.logging.MemoryHandler.push =

java.util.logging.MemoryHandler.target =