The java.util.logging.LogManager manages the internal Logger hierarchy, and initiates the configuration of the Logger's, either through the configuration class or configuration file.

There is only a single LogManager instantiated in the whole JVM. It is a singleton, in other words. Here is how to obtain the LogManager instance:

LogManager manager = LogManager.getLogManager();

You will not often need to interact directly with the LogManager, except for a few borderline cases.

For instance, if you want to reload the configuration file, you can do so using either of these methods:

readConfiguration();

readConfiguration(inputStream);

The first method on the LogManager simply re-reads the configuration from file (or class), in case these has changed.

The second method on the LogManager simply reads the configuration from the given InputStream.

You can also get access to an MXBean (Java Management Extensions) from the LogManager using the method getLoggingMXBean(). Here is an example:

LoggingMXBean mxBean = logManager.getLoggingMXBean();

The LogManager has more methods than these two, but most often you will not need them. Check the official JavaDoc for more information, if you need to do something not covered here.