Customizing the logging configuration

Overview

VIVO uses the Log4J package for logging status messages. VIVO is shipped with a configuration file that sets up the logging properties, so the VIVO log is written to `vivo.all.log` in the *`[tomcat]`*`/logs` directory. Most sites find this default configuration suitable when they start out, but often as people become more experienced with VIVO, they prefer to change the logging options.

The default configuration

The configuration file is found at *`[vitro]`*`/webapp/config/log4j.properties`. The file looks something like this:

```
log4j.appender.AllAppender=org.apache.log4j.RollingFileAppender
log4j.appender.AllAppender.File=$${catalina.home}/logs/${webapp.name}.all.log
log4j.appender.AllAppender.MaxFileSize=10MB
log4j.appender.AllAppender.MaxBackupIndex=10
log4j.appender.AllAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.AllAppender.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss,SSS} %-5p [%c{1}] %m%n

log4j.rootLogger=INFO, AllAppender

log4j.logger.edu.cornell.mannlib.vitro.webapp.startup.StartupStatus=WARN
log4j.logger.edu.cornell.mannlib.vitro.webapp.dao.jena.pellet.PelletListener=WARN
log4j.logger.org.springframework=WARN
log4j.logger.com.hp.hpl.jena.sdb.sql.SDBConnection=ERROR
```

(The listing above has been abridged for clarity. Comments have been removed, as have some repetitious lines.)

The file creates an "appender", which tells Log4J where to write the log messages, and how to manage them. It creates a "root logger" which will set the default properties for all logging: using the named appender and omitting any messages that are lower than INFO level. Finally, it overrides the logging threshold level for some special classes and packages.

In more detail (by line numbers):

> (1) Use a RollingFileAppender. This will write messages to the named file, until the file becomes too large. Then the accumulated messages are "rolled over" to a backup file, and logging continues.
>
> (2) Specify the name and location of the log file. During the build process, `${webapp.name}` will be replaced by `vivo`, or whatever you have chosen as the name of your webapp. When VIVO starts, Log4J will replace `$${catalina.home}` with the value of the system property named `catalina.home`. This is the Tomcat home directory.
>
> (3) Files will roll over when they reach 10 MegaBytes of content.
>
> (4) No more than 10 files will be kept
>
> (5, 6) The message layout is determined by this pattern. It consists of the date and time, the severity of the message, the name of the class writing the message, and the message itself (followed by a linefeed).
>
> (8) The root logger, and by default all loggers, will write to this appender. Only messages with a level of INFO or higher will be written to the log. That is, messages with levels of DEBUG or TRACE will not be written.
>
> (10, 11) Override the defaults for these classes. The write too many INFO messages, so we restrict them to WARN or higher.
>
> (12) Override the default for the entire package of `org.springframework`
>
> (13) Don't show messages from `com.hp.hpl.jena.sdb.sql.SDBConnection` unless they are ERROR or FATAL.

Writing some messages to a special log

Here is an example of how to override the defaults for particular classes in VIVO. In this example, the messages associated with rebuilding the search index are to be written to a special log file. The messages about re-inferencing are also to be written to that file.

The lines below can be added to the end of the default configuration:

```
log4j.appender.SpecialAppender=org.apache.log4j.DailyRollingFileAppender
log4j.appender.SpecialAppender.DatePattern='.'yyyy-MM-dd
log4j.appender.SpecialAppender.File=/usr/local/vivo/logs/inference_and_indexing.log
log4j.appender.SpecialAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.SpecialAppender.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss,SSS} %-5p [%c{1}] %m%n
log4j.logger.edu.cornell.mannlib.vitro.webapp.search.indexing.IndexBuilder=SpecialAppender
log4j.logger.edu.cornell.mannlib.vitro.webapp.search.indexing.IndexWorkerThread=SpecialAppender
log4j.logger.edu.cornell.mannlib.vitro.webapp.reasoner.ABoxRecomputer=SpecialAppender
```

Here we define a second appender, and tell three particular Java classes to use that appender.

Again, by line numbers:

> (15, 16) Use a DailyRollingFileAppender. Unlike the RollingFileAppender, this log file will roll over at midnight every day. There is no maximum number of files.
>
> (17) The log file will be `/usr/local/vivo/logs/inference_and_indexing.log`. At midnight, the file will be renamed to `inference_and_indexing_log.2013-06-21` (for example).
>
> (18, 19) The layout of the message is the same as for the main log file
>
> (20, 21, 22) These three classes will write to the new appender.

Notice that the log messages for these classes will now be written both to the main log file and to this special file. By default, the appenders are "added" to the classes where they are specified. If you want these classes to only write to the special file, you must turn off the "additivity" property of those classes, as shown below:

```
log4j.additivity.edu.cornell.mannlib.vitro.webapp.search.indexing.IndexBuilder=false
log4j.additivity.edu.cornell.mannlib.vitro.webapp.search.indexing.IndexWorkerThread=false
log4j.additivity.edu.cornell.mannlib.vitro.webapp.reasoner.ABoxRecomputer=false
```

More information

Log4J is a very powerful and flexible framework. Many different options are available through the use of appenders, layouts, and filters. For more information, you may want to consult

- The Log4J manual – a compact discussion of the many aspects of Log4J.
- The Log4J API documentation
- The documentation of the Log4j properties file