Events log4j Parser (events-log4j-parser) TODO

* lg help gets stuck.
* User manual that mentions -f
* Consolidate findNext() across components.
* Consolidate injectIntoLog4jEvent() into a base text.

cdel

t -c LevelTest

* WildFly %E implementation and Test
* Fix all tests.
* Re-map the io.novaordis.events.log4j.pattern package to <https://kb.novaordis.com/index.php/Log4j_Pattern_Layout> terms:
  + Remove the semantics of ConversionSpecifier. in that it may never return null, and then make it return ‘char’ not “Character”.
  + LiteralPatternElement must be NOT a ConversionSpecifier. Introduce a common interface.
* **log4j pattern.** Generalize by adding support for a log4j pattern. Currently we’re only using heuristics. <http://logging.apache.org/log4j/2.x/manual/layouts.html#PatternLayout>

cdel

t -d LevelPatternElementTest

t -c Log4jParserTest

* + Fix all tests.
  + Continue with the extension of the parser that actually uses the pattern layout, in an inverse transformation.
  + Make sure heuristics still works in absence of pattern.
  + Implement the simplest possible solution so I can parse Centric’s logs.

cd ~/tmp

lg --format='%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n' ./server.log

* Add the logic to lg to detect crossing timestamps.
* **Format instances thread safety**. Remove public static final SimpleDateFormat and other Formats everywhere, and replace them with static method that create the instances on demand. We need to do this because Format instances are not thread safe. See io.novaordis.events.api.event.DateProperty. getDefaultDateFormat() as example.
* lg rate log-level:ERROR from: to:
* **Replace long field names with shorter ones** that make sense from a log4j perspective – use the corresponding names of the log4j concepts.

timestamp -> time

line-number(Long) -> line

log-category(String) -> category

log-level(String) -> level

message(String) -> msg?

raw(String)

thread(String)

* **Field Query Sanity Check.** If we are using a field query, and none of the events query has that field, warn, because we may be using the wrong field name, not field content.
* **lims.** Write the equivalent of the shell “lims” in log4jp. The bottom limit should not be obtained via sequential processing, but we should go to the end of the file on disk and start processing from there up.
* **Compare output with grep.** Figure out this, why grep finds more events than us:

NOMBP2:03-local-sparky-2.10-3CPU ovidiu$ grep "[0-2]\\d:[0-5]\\d:[0-5]\\d,\\d\\d\\d " server.log | wc -l

84440

NOMBP2:03-local-sparky-2.10-3CPU ovidiu$ log4jp ./server.log | wc -l

78550

NOMBP2:03-local-sparky-2.10-3CPU ovidiu$