Exabeam Best Practice

**TABLE OF CONTENTS**

[Foreword 1](#_Toc20965572)

[Thought Bot's Code Review Style Guide 1](#_Toc1389449830)

[Best Practice for Everyone 1](#_Toc1625007961)

[Best Practice While Having Your Code Reviewed 2](#_Toc1712769200)

[Best Practice While Reviewing Code 2](#_Toc732899865)

[Style Comments 3](#_Toc837242296)

[Further Reading 3](#_Toc2119604221)

# Best Practice

## **Gotcha: Category and Categorization**

Different things!

## Exabeam parsers

* Parser Spec  
  <https://docs.exabeam.com/en/content/all/how-content-works-guide/56904-exabeam-parsers.html>
* Exabeam official Parsers  
  <https://github.com/ExabeamLabs/Content-Doc>
* Suspected Java Regex Engine  
  <https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>

### Exabeam Additional Parser Guidelines

<https://docs.exabeam.com/en/content/all/how-content-works-guide/56904-exabeam-parsers.html>

Here are a few very important notes to keep in mind when working with parsers:

* If time is not available in the raw log, use the syslog field headers.
* Without parsing a user, src\_ip, dest\_ip, dest\_host, or src\_host , Advanced Analytics cannot process the event and the log will be of no value to you.
* Parsers are organized into major vendors . For example, parsers for logs generated by Carbon Black products can be found in config/default/parsers\_carbonblack.conf.
* Recommended to test your regex with java8 engine on <https://regex101.com/>

## REGEXES MISPARSING

Design your regexes to be able to capture all possible variations of your data. By carefully creating and testing your regexes, you can make sure that Advanced Analytics doesn't miss data that would prevent it from being able to model a specific field.

Design your regexes to capture edge cases for how a value might appear in the log. In many cases, regexes are initially built to end when a space or an array/log-constructor-like character (':','[', '}') is used to end the regex. You will need to balance the requirements to allow a broadly tuned regex to capture what should be required, as well as limit how far the regex is allowed to capture. Sometimes a regex change is required due to a space being allowed in a filename, or the log management system happens to be appending forward or back slashes.

## PERFORMANCE TUNING

The speed of a regex is crucial for the stability of the ingestion engine. A single high volume log source that hits a single parser that takes 70 ms to parse a single log will severely degrade performance . Starting with Advanced Analytics I48, parsers that impact the ingestion process as a whole will be automatically disabled.

* generic link to common regex pitfalls  
  <https://www.loggly.com/blog/five-invaluable-techniques-to-improve-regex-performance/>

## Exabeam Expression

* Useful helpers for exabeam
* <https://community.exabeam.com/s/article/Writing-Expressions-in-Exabeam>

## Exabeam .ldf format

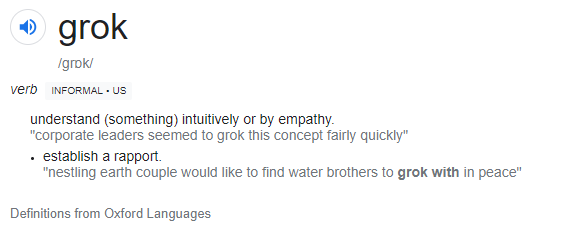
* <https://docs.fileformat.com/database/ldf/>

## Exabeam .conf format

* <https://jbrowse.org/docs/configuration_file_formats.html>

# Trivial

## Grok



<https://www.google.com/search?q=grok>   
<https://en.wikipedia.org/wiki/Grok>

# Further Reading

* official exabeam community docs
* <https://community.exabeam.com/s/>