ONTOTEXT ANSWER:

Logging in GraphDB is complex, leading to some confusion about how to best address it. Luckily, it all follows a logical structure. First, let's examine what logs are available:

- **Main log** – for operational and error messages. Contains information about your overall system state, active repositories, plugins, operations.
- **Error log** – contains only error messages. They are also present, usually with more context, in the main log. The error log is useful mostly in cases where the main log moves too fast to examine the main log.
- **Query log** – contains a log of the contents of all update queries. The idea behind the log is that all file inserts would leave a trace somewhere, unless deleted explicitly, but an update query would be invisible to anyone but the person executing it. *If you set this log to debug level, it would store all queries*.
- **Slow query log** – just like the query log, but only for queries that take more than a set amount of seconds. This is controlled by *slow-op-threshold-ms* and defaults to 60. Useful for debugging.
- **Http log** – logging for the low-level Apache HTTP client. Defaults to INFO, which would never print messages. Set to DEBUG or TRACE to check the low-level system traffic.
- **Cluster log** – this log only appears when operating in a cluster. It contains information about cluster connections and state. Some of that data duplicates the main log, but not all of it.
- **Audit log** – this log only appears when enabled. Contains information about each HTTP request – from which IP address did it come, with which user, what were the contents (truncated), what was the response code.
- **Transaction log** – not the same as all other logs, this "log" is actually a long queue of transactions that the cluster workers must process. It isn't human-readable.

Of all of these, the main log is the log that is usually examined. The cluster log is useful when operating in a cluster, but it's most valuable when used together with the main log. The transaction log is the main component of the GraphDB cluster. All other logs have very specific use cases. If you ever need to report an issue to us, do not worry, all recent logs are included in the server report.

You can control the logs via a logback.xml file. By default, it's located inside the conf

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com

directory, together with the GraphDB properties file. However, if you lose track of it, GraphDB will print out where it sources its logging configurations from in the console upon startup.

If you are operating in production, it's a good idea to set a rolling policy inside the logback, as the logs are never cleaned by default. Here's an example to get you started:

```
<rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <maxHistory>30</maxHistory>
    <fileNamePattern>${logDestinationDirectory}/enterprise-log-%d{yyyy-MM-dd}.log</fileNamePattern>
</rollingPolicy>
```

Did this help you solve your issue? Your opinion is important not only to us but also to your peers.

Help us get better: review GraphDB on Gartner

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com