USE OF THE CLUSTER LOG

(Note: none of this applies to the local “dout” logging. This is about the cluster log that we send through the mon daemons)

SEVERITY

Use ERR for situations where the cluster cannot do its job for some reason. For example: we tried to do a write, but it returned an error, or we tried to read something, but it’s corrupt so we can’t, or we scrubbed a PG but the data was inconsistent so we can’t recover.

Use WRN for incidents that the cluster can handle, but have some abnormal/negative aspect, such as a temporary degredation of service, or an unexpected internal value. For example, a metadata error that can be auto-fixed, or a slow operation.

Use INFO for ordinary cluster operations that do not indicate a fault in Ceph. It is especially important that INFO level messages are clearly worded and do not cause confusion or alarm.

FREQUENCY

It is important that messages of all severities are not excessively frequent. Consumers may be using a rotating log buffer that contains messages of all severities, so even DEBUG messages could interfere with proper display of the latest INFO messages if the DEBUG messages are too frequent.

Remember that if you have a bad state (as opposed to event), that is what health checks are for – do not spam the cluster log to indicate a continuing unhealthy state.

Do not emit cluster log messages for events that scale with the number of clients or level of activity on the system, or for events that occur regularly in normal operation. For example, it would be inappropriate to emit a INFO message about every new client that connects (scales with #clients), or to emit and INFO message about every CephFS subtree migration (occurs regularly).

LANGUAGE AND FORMATTING

(Note: these guidelines matter much less for DEBUG-level messages than
for INFO and above. Concentrate your efforts on making INFO/WRN/ERR messages as readable as possible.)

Use the passive voice. For example, use “Object xyz could not be read”, rather than “I could not read the object xyz”.

Print long/big identifiers, such as inode numbers, as hex, prefixed with an 0x so that the user can tell it is hex. We do this because the 0x makes it unambiguous (no equivalent for decimal), and because the hex form is more likely to fit on the screen.

Print size quantities as a human readable MB/GB/etc, including the unit at the end of the number. Exception: if you are specifying an offset, where precision is essential to the meaning, then you can specify the value in bytes (but print it as hex).

Make a good faith effort to fit your message on a single line. It does not have to be guaranteed, but it should at least usually be the case. That means, generally, no printing of lists unless there are only a few items in the list.

Use nouns that are meaningful to the user, and defined in the documentation. Common acronyms are OK – don’t waste screen space typing “Rados Object Gateway” instead of RGW. Do not use internal class names like “MDCache” or “Objecter”. It is okay to mention internal structures if they are the direct subject of the message, for example in a corruption, but use plain english. Example: instead of “Objecter requests” say “OSD client requests” Example: it is okay to mention internal structure in the context of “Corrupt session table” (but don’t say “Corrupt SessionTable”)

Where possible, describe the consequence for system availability, rather than only describing the underlying state. For example, rather than saying “MDS myfs.0 is replaying”, say that “myfs is degraded, waiting for myfs.0 to finish starting”.

While common acronyms are fine, don’t randomly truncate words. It’s not “dir ino”, it’s “directory inode”.

If you’re logging something that “should never happen”, i.e. a situation where it would be an assertion, but we’re helpfully not crashing, then make that clear in the language – this is probably not a situation that the user can remediate themselves.

Avoid UNIX/programmer jargon. Instead of “errno”, just say “error” (or preferably give something more descriptive than the number!)

Do not mention cluster map epochs unless they are essential to the meaning of the message. For example, “OSDMap epoch 123 is corrupt” would be okay (the epoch is the point of the message), but saying “OSD 123 is down in OSDMap epoch 456” would not be (the osdmap and epoch concepts are an implementation detail, the down-ness of the OSD is the real message). Feel free to send additional detail to the daemon’s local log (via *dout*/*derr*).

If you log a problem that may go away in the future, make sure you also log when it goes away. Whatever priority you logged the original message at, log the “going away” message at INFO.