

**git\_comments:**

1. \* \* Get a friendly description of the error (if one is available). \* @return the error message

**git\_commits:**

1. **summary:** KAFKA-3318: clean up consumer logging and error messages  
**message:** KAFKA-3318: clean up consumer logging and error messages Author: Jason Gustafson <jason@confluent.io> Reviewers: Ismael Juma Closes #1036 from hachikuji/KAFKA-3318  
**label:** code-design

**github\_issues:**

**github\_issues\_comments:**

**github\_pulls:**

1. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
2. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
3. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
4. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
5. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
6. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
7. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** documentation
8. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
9. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
10. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
11. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
12. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
13. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
14. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
15. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
16. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design

17. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
18. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
19. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
20. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
21. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
22. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
23. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
24. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
25. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
26. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
27. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
28. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
29. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
30. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
31. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
32. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
33. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
34. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
35. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
36. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
37. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**
38. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
39. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
40. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design
41. **title:** KAFKA-3318: clean up consumer logging and error messages  
**body:**  
**label:** code-design

## github\_pulls\_comments:

1. **body:** Minor comment, otherwise LGTM!  
**label:** code-design
2. **body:** Thanks for this @hachikuji, very important. :) I left a few comments, looks good otherwise.  
**label:** code-design
3. **body:** @ijuma Made some more improvements. In particular, I wanted to ensure that the groupId was always provided since otherwise it's difficult to follow logs which contain multiple consumer instances. I also tried to address the review comments.  
**label:** code-design
4. @hachikuji I did another pass. That last commit changed more than the previous ones! I think it's looking pretty good now. I had a small number of comments/questions.
5. LGTM

## github\_pulls\_reviews:

1. **body:** Typo "poll() by with max.poll.records"?  
**label:** documentation
2. Ack.
3. **body:** Do we really need two log statements so near to each other? I guess the second one also includes the co-ordinator id. Would it be too confusing to include it in the previous log statement?  
**label:** code-design
4. Where do the retries happen and what message do we log when all the retries fail?
5. **body:** I guess we also need this debug log because of the `syncResponse`? Does the message still make sense given the new log statement above it?  
**label:** code-design
6. Would this read better as "Not authorized to..."?
7. **body:** Should this be `error.message()` like a few lines above? Same question for other cases where we are still using `error`.  
**label:** code-design
8. **body:** Do we want to say "most likely means" or simply "may mean"? There could be other reasons too, right? Do we want to briefly mention them?  
**label:** code-design
9. What is the reasoning for this being a warning?
10. **body:** Should the second sentence be "Will find new coordinator and retry"? That seems to be what we use everywhere else.  
**label:** code-design
11. **body:** The coordinator id ends up being a really big negative integer, but if we used host:port in the log message then maybe it wouldn't be too bad? We'd probably want to make the same change in the other messages as well.  
**label:** code-design
12. **body:** It happens elsewhere. We could probably just leave that part off if we're reducing the level to debug. Seems like it was originally intended to give users some comfort that the error is being handled internally.  
**label:** code-design
13. Sure.
14. I was debating on it. Maybe there's some useful information to pull out of the sync response, but probably not if it was successful anyway. Perhaps we just remove it?
15. Ack.
16. Interesting. Which is more useful host:port or id? If the latter, let's just leave as is. Otherwise, then changing it here and other places sounds good.
17. Ack.
18. I changed it to warn because we ignore it internally.
19. How about this? > This means that the time between subsequent calls to poll() was longer than the configured session.timeout.ms, which typically means that the poll loop is spending too much time message processing.
20. Usually the host:port is more helpful when debugging, and it's kind of a pain to do the reverse mapping from id to host in my head. The only time the id is helpful is when looking at NetworkClient issues.
21. Sounds good.

22. **body:** Should we change the `toString` of `Node` to be more logging friendly?  
**label:** code-design
23. Yeah, it seems to me like we should remove it.
24. **body:** Is it better to say `Unexpected error` since we are not actually outputting the `errorCode` in the message?  
**label:** code-design
25. I think we tend to just say `partition` even though we mean `topic partition`. Is there a reason to say `topic-partition` here?
26. But we want users to be aware of it?
27. Only because it's an unexpected error code. I could go either way.
28. **body:** Good point. We probably `_should_` include the error code or it will be tough to debug the issue.  
**label:** code-design
29. Ack.
30. OK, let's leave it as warn.
31. It currently looks like `Node(id, host, port),` which doesn't seem too bad to me. What do you think?
32. **body:** When I suggested it, I thought we could do a bit better, maybe something like `(id: 5, www.example.com:9123)`, but maybe that's actually worse.  
**label:** code-design
33. **body:** I was mostly trying to get rid of the word `Node` because it's a bit redundant when you look at the log messages.  
**label:** code-design
34. **body:** Initially I didn't use `Node's toString()` at all. I just filled in `"{}: {}"` with the host and port, but that got a little annoying after a couple times. But maybe it's actually better?  
**label:** code-design

#### jira\_issues:

1. **summary:** Improve consumer rebalance error messaging  
**description:** A common problem with the new consumer is to have message processing take longer than the session timeout, causing an unexpected rebalance. Unfortunately, when this happens, the error messages are often cryptic (e.g. something about illegal generation) and contain no clear advice on what to do (e.g. increase session timeout). We should do a pass on error messages to ensure that users receive clear guidance on the problem and possible solutions.
2. **summary:** Improve consumer rebalance error messaging  
**description:** A common problem with the new consumer is to have message processing take longer than the session timeout, causing an unexpected rebalance. Unfortunately, when this happens, the error messages are often cryptic (e.g. something about illegal generation) and contain no clear advice on what to do (e.g. increase session timeout). We should do a pass on error messages to ensure that users receive clear guidance on the problem and possible solutions.  
**label:** code-design
3. **summary:** Improve consumer rebalance error messaging  
**description:** A common problem with the new consumer is to have message processing take longer than the session timeout, causing an unexpected rebalance. Unfortunately, when this happens, the error messages are often cryptic (e.g. something about illegal generation) and contain no clear advice on what to do (e.g. increase session timeout). We should do a pass on error messages to ensure that users receive clear guidance on the problem and possible solutions.

#### jira\_issues\_comments:

1. GitHub user hachikuji opened a pull request: <https://github.com/apache/kafka/pull/1036> KAFKA-3318: clean up consumer logging and error messages You can merge this pull request into a Git repository by running: `$ git pull https://github.com/hachikuji/kafka KAFKA-3318` Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/kafka/pull/1036.patch> To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #1036 ---- commit 274378395e48bca11350c1df3a38137810ba0732 Author: Jason Gustafson <jason@confluent.io> Date: 2016-03-09T01:42:11Z KAFKA-3318: clean up consumer logging and error messages ----
2. Issue resolved by pull request 1036 [<https://github.com/apache/kafka/pull/1036>]
3. Github user asfgit closed the pull request at: <https://github.com/apache/kafka/pull/1036>

