



Toolforge webservices are in the final stages of [migrating to the toolforge.org domain](#).  
Please help us clean up older documentation referring to [tools.wmflabs.org](#)!

# Incident documentation/20180705-Eventlogging-in-Hive

[< Incident documentation](#)

## Contents [\[hide\]](#)

- [1 Summary](#)
- [2 Timeline](#)
- [3 Conclusions](#)
- [4 Links to relevant documentation](#)
- [5 Actionables](#)

## Summary

On July 5, Marcel noticed that there were missing hourly partitions from refined EventLogging tables in the Hive 'event' database. The problem was caused by an errant log segment deletion by Kafka, which deleted data too early. This caused Camus to fail importing data for the affected topics, as it could not resume from its stored offset, as the offset no longer existed. We have monitoring to check that Camus imported data is successfully refined, but no monitoring of EventLogging Camus jobs or imports, so we did not notice that this happened until weeks later. Fortunately, most of the affected events are low volume and also exist in MySQL and/or in the archived all-events.log files we keep, so backfilling was possible.

## Timeline

- Thursday July 5: Marcel notices missing EventLogging Hive table partitions and notifies Otto.
- Thursday July 5: Otto investigates, and finds errors in Camus logs about not being able to consume due to offset out of range error.
- Thursday July 5: Otto changes camus settings: `kafka.move.to.earliest.offset=true` and `kafka.max.historical.days=-1` and re-runs Camus for EventLogging. This causes Camus to reset its offset from the earliest available in Kafka (7 days prior), and Camus re-imports the previous 7 days.
- Thursday, July 5: Otto greps for affected topics in Camus logs, and then uses this list to [grep for missing events](#) in the all-events.log on eventlog1002.
- Thursday July 5: Otto then [replays these events through eventlogging-consumer](#) so that they flow to their destined Kafka topics.
- Friday July 6: Once all events have been replayed and Camus has imported and time hourly bucketed the old events, [Otto launches a backfill Spark Refine job](#). However, he mistakenly used an out-dated json-refine wrapper script (that should have been manually deleted when it was removed from Puppet) to do this. This old job does not deduplicate or geocode events. Some partitions were refined before Otto realized his mistake and stopped the job. Now since some of the affected partitions were successfully refined, we decided to remove all `_REFINED` done flags from affected table partitions in the date range to force re-refinement of all this data.
- Friday July 6: Otto then re-launches the [proper backfill Spark Refine job](#) waits for it to finish.
- Sunday July 8: Otto notices that the job has finished. He grabs the YARN application logs (16G worth!) and greps them for [relevant status log lines](#).
- Monday July 9: Otto and Marcel do a rough spot check to verify that the expected Hive partitions now exist and are re-refined.

## Conclusions

We need better monitoring of Camus if we are going to continue to use it for event data imports into Hadoop. We have great monitoring around the webrequest data, but the event schemas are too numerous to manually monitor in the same way. We need some short term spot checking monitoring (manual monitoring of a couple of canary

[Main page](#)  
[Recent changes](#)  
[Server admin log \(Prod\)](#)  
[Server admin log \(RelEng\)](#)  
[Deployments](#)  
[SRE/Operations Help](#)  
[Incident status](#)

[Cloud VPS & Toolforge](#)  
[Cloud VPS documentation](#)  
[Toolforge documentation](#)  
[Request Cloud VPS project](#)  
[Server admin log \(Cloud VPS\)](#)

[Tools](#)  
[What links here](#)  
[Related changes](#)  
[Special pages](#)  
[Permanent link](#)  
[Page information](#)  
[Cite this page](#)

[Print/export](#)  
[Create a book](#)  
[Download as PDF](#)  
[Printable version](#)

topics), and in the long term better automated event/topic auditing.

I don't expect this exact problem to happen again. I believe it was caused by a one time Kafka broker config change when we changed `log.message.timestamp.type` from `LogAppendTime` to `CreateTime` to take advantage of producer set event timestamps. We won't be changing this setting again.

## Links to relevant documentation

---

- <https://phabricator.wikimedia.org/T198906>↗
- [Analytics/Systems/EventLogging/Backfilling](#)

## Actionables

---

- [phab:T198908](#)

Category: [Incident documentation](#)

This page was last edited on 19 October 2018, at 12:06.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. See [Terms of Use](#) for details.

[Privacy policy](#) [About](#)  
[Wikitech](#)

[Disclaimers](#) [Code of Conduct](#) [Developers](#) [Statistics](#) [Cookie statement](#) [Mobile view](#)

