



Toolforge webservises 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/20180306-MasterMergedInWMFBranch

[< Incident documentation](#)

Contents [\[hide\]](#)

- [1 Summary](#)
- [2 Timeline](#)
- [3 Conclusions](#)
- [4 Actionables](#)

Summary

During deployment of a patch for mediawiki/core@wmf/1.31.0-wmf.23, the deployer found out the difference between HEAD the origin/wmf/1.31.0-wmf.23 had too many commits. That prompted for an immediate halt.

The root cause has been determined to be the commit from master being sent for review to the wmf branch. In such a case, Gerrit will **implicitly merge master into wmf branch** . The Gerrit UI was not showing that and when the patch got merged the master branch made it to the wmf branch.

Timeline

- Mar 05 - CR [416499](#) for mediawiki-core submitted and merged into the master branch
- Mar 06 13:18 UTC - The change cherry-picked from the master branch to wmf/1.31.0-wmf.23 branch with CR [416675](#) and merged. Gerrit implicitly merged all the ancestors of the commit from master branch to wmf/1.31.0-wm.23 branch
- Mar 06 13:18 UTC Attempt to deploy revealed that the diff is too big and deploy was halted.
- Mar 06 14:46 UTC A revert of the erroneous commit created [416686](#) and merged, the state of gerrit returns back to normal.

Conclusions

- Deployer followed the documentation (`git diff HEAD..HEAD@{u}`) and properly halted on the first oddity
- Gerrit is being too helpful

The reason behind is:

Given a git tree such as:

```
A -- B -- C (master)
 \
  \-- X (wmf)
```


One is locally on the master branch and does a `git push HEAD:refs/publish/wmf`. Gerrit will create a new change Y that implicitly merge master into the wmf branch:

```
A -- B -- C (master)
 \           \
  \-- X (wmf) --\-- Z (new change)
```

The Gerrit GUI fails to reflect that the new change (Z) is a merge commit. One can tell though since the parent commit will be B from the master branch.

When Z is submitted the master branch ends up having been merged in wmf since B and C got already merged in master.

[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](#)

Gerrit 2.13 offers the option `receive.rejectImplicitMerges` to prevent that implicit merge. From [I Gerrit 2.13 release notes](#) 

The 'Reject Implicit Merges' option can be enabled to prevent non-merge commits from implicitly bringing unwanted changes into a branch. This can happen for example when a commit is made based on one branch but is mistakenly pushed to another, for example based on refs/heads/master but pushed to refs/for/stable.

Actionables

- In Gerrit disable `receive.rejectImplicitMerges` <https://gerrit.wikimedia.org/r/Documentation/config-project-config.html#receive.rejectImplicitMerges>  [phab:T189024](#)
 - Done via [gerrit:416704](#)

Category: [Incident documentation](#)

This page was last edited on 19 October 2018, at 11:56.

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

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

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

[Wikitech](#)

