

Quassel IRC - Bug #1435

Quassel client segfaults on every attempt to display a buffer with a carriage return in the topic

04/06/2017 02:25 AM - confluence

| | | | |
|------------------------|----------------|------------------------|------------|
| Status: | New | Start date: | 04/06/2017 |
| Priority: | Normal | Due date: | |
| Assignee: | | % Done: | 0% |
| Category: | Quassel Client | Estimated time: | 0.00 hour |
| Target version: | | OS: | Linux |
| Version: | 0.12.4 | | |

Description

I encountered this issue when connecting to a Slack team through an IRC gateway. I assume that the root cause is Windows users pasting multiline topics into Slack channels. I don't know if it's possible to insert a carriage return into a topic on a normal IRC server.

Steps to reproduce:

1. Connect to a Slack team's IRC gateway
2. Join or create a channel through the Slack web interface
3. Insert a carriage return into the channel's topic through the Slack web interface (it's impossible to type a multiline topic directly, but a multiline topic can be pasted in)
4. Attempt to access the channel's buffer in Quassel

I have found no workaround other than entering the affected channel through the Slack web interface and re-pasting the corrected topic (from Linux, minus the carriage return).

Is carriage return not a legal character in IRC channel topics? Should Quassel check for it and strip it out? Is it Slack's responsibility to fix this?

History

#1 - 04/06/2017 03:06 AM - confluence

I spoke too soon about the workaround.

Immediately after I change the topic (through the web), I am able to interact with the buffer normally in Quassel, but only until I disconnect from the gateway and reconnect. When I do, all multiline topics have carriage returns in them again. So the workaround is very temporary.

I considered that Slack might be ignoring topic changes if they differ only by whitespace or non-printable characters, but that does not seem to be the case. I can reproduce this even by pasting in a completely new multiline topic (typed out in an editor on Linux, definitely without carriage returns). So it looks like something adds carriage returns to all newlines in topics, but only when they are fetched after Quassel reconnects to the gateway; not when the topic is updated while Quassel is connected.

I have tried to connect to the gateway through a different IRC client (hexchat). I don't experience any issues with these buffers in that client -- everything works normally.

(Possibly related: I am also unable to merge buffers within this server -- attempting to click and drag on a buffer name just selects multiple buffers. I can merge buffers in every other Slack team I'm connected to. Same underlying cause? Completely separate issue?)

#2 - 04/06/2017 10:07 AM - confluence

Further observations:

1. I can also trigger the restoration of the carriage returns when I disconnect from and reconnect to that specific channel in Slack.
2. I can view the buffer when I am not connected to the channel -- either because I have disconnected from the gateway entirely or just left the channel in Slack. When I do this I can see the carriage returns in the topic as it is printed in the server messages in the chat view (displayed as the unicode carriage return glyph -- when I look at the debug output in the console, they're just displayed as `\r`).

Perhaps the segfault occurs when Quassel attempts to display the topic in the topic widget (which is left blank when the channel is inactive)?

#3 - 04/06/2017 11:18 AM - confluence

Last thing: please ignore my comment about not being able to merge buffers. I see now that in all my Slack gateway connections I can merge private chat buffers but not channel buffers, so this behaviour is consistent.

#4 - 04/15/2017 03:09 AM - genius3000

Just to note it here:

- The Carriage Return in topic issue has been fixed in the git master branch, when building with Qt5. Qt4 doesn't have the Carriage Return character, but perhaps another solution could be done for the 0.12 branch yet.

- Merging channel buffers has also been implemented in the git master branch

That being said, 0.12 still remains the stable branch (0.12.4 being the current stable release) and I am not recommending the use of the master branch unless you wish to accept all risks involved in using it.

#5 - 07/21/2017 02:41 PM - confluence

Thanks for the update! I somehow missed it until today. I have built the Git master branch from source with Qt 5, and everything seems to be working.