



Unify `ref-filter` formats with other `--pretty` formats

Name	:	Hariom Verma
Major	:	Computer Science and Engineering
Mobile no.	:	+91 9179078061
Email	:	hariom18599@gmail.com
IRC	:	harry-hov (on #git-devel@freenode)
Github	:	harry-hov
Linkedin	:	https://www.linkedin.com/in/harry-hov/
Time Zone	:	IST (UTC +05:30)

About me



I am a Computer Science undergraduate at Shri G S Institute of Technology & Science, Indore, India. I am passionate about software freedom and open source communities. I've been writing in C, C++, Python & Dart for almost 3 years, and contributed to some projects which can be seen on [Github](<https://github.com/harry-hov>).

I'm contributing to Git since Oct 2019 and would love to be a long-term member of such a versatile community.

PS: I love reading manga too.

Me & Git

My contributions have helped me understand the review and patch submission process in a better way. Also, I now feel a bit more comfortable with Git's codebase.

	Commits	Status
	<ul style="list-style-type: none">• [PATCH] t2402: test worktree path when called in .git directory	Merged into master
	Mailing List: https://public-inbox.org/git/pull.570.git.1583305200143.gitgitgadget@gmail.com/	
	Github: https://github.com/git/git/commit/4d864895a23bbb5403d9aa8a65d0576d2029597	
	<ul style="list-style-type: none">• [PATCH v3 3/3] receive.denyCurrentBranch: respect all worktrees• [PATCH v3 2/3] t5509: use a bare repository for test push target• [PATCH v3 1/3] get_main_worktree(): allow it to be called in the Git directory[3]	Merged into master
	Mailing List: [3/3]: https://public-inbox.org/git/d21a590d6c23f231c54b731b737c363b83660f79.1582484231.git.gitgitgadget@gmail.com/ [2/3]: https://public-inbox.org/git/ae749310f067c43429741987cd9f47c1ae4ceb3f.1582484231.git.gitgitgadget@gmail.com/ [1/3]: https://public-inbox.org/git/8718facbc951614f19407afa6ca8d6110507483d.1582484231.git.gitgitgadget@gmail.com/	
	Github: [3/3]: https://github.com/git/git/commit/4ef346482d6d5748861c1aa9d56712e847369b40 [2/3]: https://github.com/git/git/commit/f8692114dbb1b3ffe0b71871a015c632c195b784 [1/3]: https://github.com/git/git/commit/45f274fbb118cc4cb00640c30b4e3069d96755a0	



- [PATCH 1/1] git-compat-util.h: drop the `PRIuMAX` and other fallback definitions **Merged Into master**

Mailing List:

<https://public-inbox.org/git/177deddcf83c2550c0db536a7a6942ba69a92fa5.1574600963.git.gitgitgadget@gmail.com/>

Github:

<https://github.com/git/git/commit/e547e5a89e3641bae02e762bcb0062951bb698e5>



- [PATCH v2 1/1] builtin/blame.c: constants into bit shift format **Merged into master**

Mailing List:

<https://public-inbox.org/git/838478a185322572ed4747eb484b678a8e2d7ac6.1571334411.git.gitgitgadget@gmail.com/>

Github:

<https://github.com/git/git/commit/86795774bb9ca3c63b94d3d0930405c1ba9148ec>

Proposed Project

- **Abstract**

Git has an old problem of duplicated implementations of some logic. For example, Git had at least 4 different implementations to format command output for different commands. The foremost aim of this project is to simplify codebase by getting rid of duplication of a similar logic and, as a result, simplify adding new functionality. The current task is to reuse ref-filter formatting logic in pretty to minimize code duplication and to have one unified interface to extract all needed data from the object and to print it properly.

- **Previous Work**

Olga<olyatelezhnaya@gmail.com> has done great work in “Unifying Git’s format languages” during Outreachy Round 15 and continued even after that [from 28-09-2017 to 04-04-2019]. Her work is mostly related to `cat-file` and `ref-filter`.

She already did a pretty nice job in preparing ref-filter for more general usage of its formatting logic. It will give me the possibility to make a migration of pretty.c easier.

- **Potential Problem**

The biggest challenge is to ***maintain backward compatibility*** while using reuse ref-filter formatting logic in pretty. One can't just delete some commands, rewrite them and change their interface.

Also, bugs could get introduced while rewriting pretty.c. Getting rid of them is also an imperative task. So I may need to have a good knowledge of GDB.

Nit: Almost a week before the end of the internship, Olga needed to redesign the solution using a new approach(maybe to keep the logic same). I need to learn from that change in approach and will also need to take advantage of that insight while designing the solution for pretty.c

- **The Plan**

My task is to change formatting logic in pretty.c. Actually, to remove the current formatting solution and reuse solution from ref-filter (maybe without changing pretty.c logically).

As Thomas<t.gummerer@gmail.com>

[suggested](<https://public-inbox.org/git/20190331184930.GV32487@hank.intra.tgummerer.com/>), I'll start by first looking at what actually needed to be replaced (for example by studying the PRETTY FORMATS section in 'man git-log', what which verbs you can use in the ref-filter ('man git-for-each-ref') to achieve the same thing.

Then I'll research how one format is implemented in 'pretty.c', and see how a similar thing using the ref-filter is implemented in 'ref-filter.c'.

e.g. the "%(objectname:short) %(contents:subject)" (this is missing coloring, but just to get the idea) is similar to '--pretty=oneline'. Now one can try to find how 'oneline' is implemented in 'pretty.c', and could translate that to use the 'ref-filter' API.

The idea is to use various atoms present in ref-filter to implement pretty format options.

Valid atoms in ref-filter:

<https://github.com/git/git/blob/274b9cc25322d9ee79aa8e6d4e86f0ffe5ced925/ref-filter.c#L474-L520>

Also, I have gone through solid part of [Git Internals](<https://git-scm.com/book/en/v2/Git-Internals-Git-Objects>) to have an understanding of objects and refs.

Possible Approach: I can just brute-force the process, merge pretty.h and ref-filter.h, and then step-by-step clean it up. Or I can create a blank file and put there needed functions step by step. Or I can just pick one format option at a time and convert it to use ref-filter logic. I am thinking about it, still don't know what is better.

Maybe later one? (which is also used by Olga)

I guess we will sort this out ;)

Estimated Timeline

I have my end semester exams between 20 April - 5 May(tentative due to Covid-19 outbreak). I'll be a bit inactive from 10 April to 5 May for my exam preparation and submissions, which is way off the timeline.

And also I have been shortlisted for Worlds Biggest Hackathon i.e Smart India Hackathon(SIH)(<https://www.sih.gov.in/>)(36 hr Hackathon)(18 & 19 July).

I believe I have enough fuel to get started on my goals - as a result to my involvement with the organisation for almost 5 months now.

Period	Tasks
Inactive Period <i>April 10, 2020 - May 5, 2020</i>	<ul style="list-style-type: none">• End-Semester will be near so I'll be bit inactive, though always available on email or any other means of communication.
Community Bonding <i>May 4, 2020 - June 1, 2020</i>	<ul style="list-style-type: none">• understanding all the logic of pretty.* and ref-filter.* (what functions are used and how all formatting process is going)• Working with mentors and identifying the best candidates to be converted first.• rewriting pretty.c skeleton, so that it could reuse formatting logic from ref-filter• Converting a couple of formatting options to reuse ref-filter formatting logic.• Tweaking ref-filter. So its logic can be reused in pretty.• Update Documentation.
My BirthDay (during Community Bonding) <i>May 18, 2020</i>	<ul style="list-style-type: none">• Typical cake-cutting ceremony.
Coding phase 1 <i>June 1, 2020 - July 3, 2020</i>	<ul style="list-style-type: none">• Tweaking ref-filter for more general usage.• Converting more formatting options to reuse ref-filter formatting logic.• Update Documentation.
Coding phase 2 <i>July 3, 2020 - July 31, 2020</i>	<ul style="list-style-type: none">• Converting remaining formatting options to reuse ref-filter formatting logic.• Teach new formatting options to --pretty using ref-filter• Update Documentation.
Final Coding Phase <i>July 31, 2020 - Aug 24, 2020</i>	<ul style="list-style-type: none">• Final touch-ups and bug fixes(if any)• Update Documentation• Wrapping up.

*Debugging and improving the test coverage will always be the priority in every phase.

*During May 4, 2020 - August 24, 2020, a blog series (<https://harry-hov.github.io/blogs/>) will be written on a weekly basis about my current findings and will also write about “What I’m working on” to further provide the status of my summer project.

Blogging about Git

I have always been reluctant to articulate my opinions and broadcast them to the world in any way. I have a long way to go as a writer, thinker, and as a human being before my opinions on matters of personal significance to me will be worth sharing. (Whoa! I have opinions about my opinions! Ad infinitum!)

Having said that, I think GSoC and Git might be a great couple for facilitating my first foray into technical blogging. And over the course of the summer and beyond, I would love to write about my progress and experiences contributing to Git.

Availability

My exams will end on 5 May. I might travel home after that. So, I can start actively writing code for this project after 7 May. My college resumes on 15 July. I can easily devote 45-50 hours a week until my college reopens and 35-40 hours per week after that. I'm also free on the weekends and I intend to complete most of the work before my college reopens. Other than this project, I have no commitments for this summer. I will also periodically post my status through blogs.

Post GSoC

As I said, I would love to be a member of Git’s community. So, I intend to keep contributing even after the GSoC period ends.

Here are some other things I’d like to do beyond GSoC

- I’d love to co-mentor someone for Git (someday!)
- I just can’t ignore my love for Algorithms, so most probably after GSoC I would like to work on “[Accelerate rename detection and range-diff](#)”.
- I love writing, and I’d love to work on improving Git’s documentation and work on documentation issues.
- Eventually, I’d like to learn more about git-for-windows and contribute to git-for-windows code base
- Keep learning from all of you...

Experience with Open Source

- Mentor, Kharagpur Winter of Code (KWoC)
Kharagpur Winter of Code is a 5-week long online program for students who are new to Open Source Development.
KWoC 2019(Mentor):
<https://kossiitkqp.org/public-files/KWoC/2019-Certificates/KWoC19-Hariom%20Verma.pdf>
KWoC 2018(Participant):
<https://kossiitkqp.org/public-files/KWoC/2018-Certificates/KWoC18-Hariom%20Verma.pdf>

Motivation

Git is everyone's favorite VCS. Apart from that... once I made some initial bite-sized contributions to git, I really loved how easy it was to get quick feedback from the community.

Closing remarks (Optional)

I am a quick learner, and always willing to put in the extra effort to cultivate the requisite skills needed to solve a given problem, even if I don't possess said skills to begin with!

I feel like my way of looking at problems has served me well while contributing to Git and various open source projects.

I hope you will give me a chance to prove that by considering my application for working with you over the summer on my proposed project! Really looking forward to learning from all of you! Here's to a summer full of learning and growth!

Heartfelt thanks and warm regards!

Hariom Verma

Proposal review thread on Mailing list

- [v1] https://public-inbox.org/git/CA+CkUQ9Pnta8Ut9ZaeKLSPn_FyQwh=Wgm1eiSWirghgtcWZqoA@mail.gmail.com/
[v2] https://public-inbox.org/git/CA+CkUQ-dhyd_5ey1z-v59FzNyWFFVosMBfunSnN5t1EqxVMvVw@mail.gmail.com/
[v3] https://public-inbox.org/git/CA+CkUQ_JPsLCkddHg7gh8J7VvXLA+v2j3NgV6cZAw94if=9mA@mail.gmail.com/
[v4] <https://public-inbox.org/git/CA+CkUQ966swTrR7D2vxgQ2ZA3E=Le=u8yvEAopOspHoCWGgDeg@mail.gmail.com/>
-