



# GSoC 2023: Unify ref-filter format with other pretty formats

---

## Personal Details

Name	Vinayak Dev
Email	vinayakdev.sci@gmail.com
Github	<a href="https://github.com/vinayakdsci">https://github.com/vinayakdsci</a>
Country of Residence	India
Education	Bachelors of Technology in Computer Science and Engineering
Institute	University of Engineering and Technology, Roorkee
Ph. No	+91-9119055909

## Overview/Synopsis

Git has had the problem of different implementations to format command outputs for different commands in the past. Through the efforts of the community members and the past GSoC and Outreachy participants, much of the formatting logic has been unified into `ref-filter.c`, allowing the code base to get rid of duplicate implementations.

The current task of the project is to continue this effort and use the formatting logic from `ref-filter.c` in `pretty.c`, to allow getting rid of duplicate formatting logic.

Outreachy participant Olga Telezhnaya and GSoC participant Hariom Verma provide a good insight into the logic and approach required for using `ref-filter` formatting logic with other commands, like `git cat-file`, etc. Olga also makes an attempt to generalize the formatting logic in `ref-filter.c` to allow for migration to other commands easier.

Their blogs at [1] and [2] are a good starting point to understand the structure and the history of the work on this idea.

Nsengiyumva Wilberforce also sent patches to the mailing list[3] where he adds a new ``signature`` atom to `ref-filter.c`. He was unable to fulfill the requirements for the Outreachy programme, but sent the patches nonetheless. His work allows to get rid of the implementation of the ``signature`` atom in `pretty.c`, moving it to `ref-filter.c` for better unification of the logic with other commands.

Also, the discussion[4] on the mailing list, with a patch series concerning `pretty.c` by Anders Waldenborg add and improve upon the

features of git's pretty-formatting, with insightful comments about the process from the reviewers, Eric Sunshine, Junio C Hamano, and Jeff King.

[1] <https://medium.com/@olyatelezhnaya>

[2] <https://harry-hov.github.io/blogs/posts/>

[3] <https://lore.kernel.org/git/?q=f%3A+Nsengiyumva+Wilberforce>

[4] <https://lore.kernel.org/git/20181104152232.20671-1-anders@0x63.nu/T/#u>

## Before GSoC

I joined the Git community in February 2023, and have since then been hugely inspired by the efficient and strong workflow, and how the members encourage newcomer's contributions, despite Git being such a large scale and important project.

To understand the source code, I have referred to many sources, some of them being the Reference Manual, Git Internals, Hacking Git, and the Pro Git book by Scott Chacon, all of which provide a good and deep insight into the working of a git, both on the level of the terminal commands, and the source code. (including MyFirstObjectWalk)

I have submitted following patches to Git:

1. [PATCH] t9160: Change test `-(d | f)` to `test_path_is_*`

Description: The patch was intended to be a microproject for Git for GSoC.

It replaces the `test -f` and `test -d` commands in `t/t9160` to helper functions described in `t/test-lib-functions.sh`, which provide better diagnostic, and error messages

Status: Discontinued

2. [PATCH v3] apply: Change #define to enum and variable types from int to enum

Description: The patch changed preprocessor constants defined in apply.c to enums, along with changing the types of variables to the respective enum types, to allow for easier debugging in case of errors (many debuggers support listing out the constants, if they are enums).

Status: WIP

## Benefits to the Community

Git is a great and big community, where people volunteer to maintain one of the best and well-written codebase of all open-source projects. I, too, aspire to be a permanent part of the community, and leave a lasting effect on the developer community. Since I joined the mailing list, I have loved the way people communicate and work together here.

So, I want remain a part of the Git community irrespective of if I am able to make it as a GSoC contributor or not, as I am sure that Git will help me become a better programmer, and a thinker in general.

This project is important as in my viewpoint removing duplicate logic and unifying it helps improve the modularity of the code base, and make it sleeker, which is one of the best qualities of the Git code base.

The project would help newcomers understand code easily, and remove scope for error-prone or failing code, which often results from duplicate logic.

## Previous and Related Work

GSoC candidate Hariom Verma and Outreachy student Olga Telezhnaya did important work in the direction of unifying formatting logic into `ref-filter.c` and removing duplicate implementations in other commands. Most importantly, Hariom's work, where he tries to create a transition table(`pretty-lib.c`) between `pretty.c` and `ref-filter.c` is important and highly relevant to this project. It provides a good starting point and gives an insight into the code in `pretty.c` and `ref-filter.c`.

His blogs(mentioned above) provide a peek into the way of thinking he adopts, as he implements the logic. This helps to understand the way of thinking of another person, which again provides knowledge.

Olga's work is important as it describes how the student might face difficulties, especially during bug fixing. This makes me confident and less hesitant to ask on the mailing list and to the mentors. Also, she describes how she generalizes the logic of `ref-filter.c` to allow for better integration to other commands, like `git cat-file`. The blog posts also emphasize the need of comfort with a debugger like GDB, but that is alright.

Hariom has clearly described, and given a link to his commits, how he adds new formatting specifiers in `pretty-formatting`.

## Proposed Work

As Christian Couder mentions on the mailing list, one of the most important parts, while removing duplicate logic in `pretty.c` is to ensure that `ref-filter.c` has to support all the atoms that `pretty-formats` have.

This is in accordance with the work of Nsengiyumva Wilberforce, who outside of Outreachy has sent patches to implement the `signature` atom in `ref-filter.c`. This patch-series is therefore an important reference for much of the work in this project.

Therefore the target/proposed work is to implement at least some of the atoms supported by `pretty-formats` into `ref-filter.c`, and then clean up the duplicate logic. This should provide a neat and clean way of working when applied incrementally.

## Timeline

April 5 - May 4 - Contribute small but meaningful patches regularly, as this is the best way to get comfortable with the source code. Also, this would give me an opportunity to interact freely with the mentors and reviewers on the mailing list.

May 4 - May 28 - (Community Bonding) This is one of the most important phases for any GSoC contributor, as interaction with mentors and reading the documentation is as important as implementing the code for the idea. During this,

I intend to start looking for duplicate implementations in the source code for pretty formatting, that must be moved to `ref-filter.c`, along with unsupported atoms.

May 29 - July 13 - The target is to re-implement the duplicate logic, but it is important to keep track of time. so, (after reading Hariom's and Olga's blogs) the best approach is to divide the work into weeks with coding, testing, and bug-fixing on specific week-days. This would help maintain a good speed, while still ensuring that the code is properly written. Also, pretty formats might have some atoms that are not supported by `ref-filter`, so begin by implementing some of them, and cleaning up the related logic.

July 14 - August 21-

Final work period involves wrapping up the code, and writing extra tests to make sure that no new errors are introduced into the code base. Also, it must be made sure that none of the old functionality breaks, so all the original tests have to pass, too.

August 22 - August 28-

This is the last week of the journey of GSoC, and the most important part is to ensure that the changes that have been made to the code base integrate well with the existing code. I hope to write good-enough code to allow for this phase to finish seamlessly.

## Blogging

Like many participants on this mailing list, and programmers in general I also find the idea of blogging about new topics and recent findings to be an extremely enriching process. However, till now, I was just playing with the idea of starting a blog, but now I am confident enough to go ahead and start one. I intend to host it on github, and post new information and updates about my progress regularly(weekly seems best). Many people I know have progressed incredibly through blogging and sharing, and I take huge inspiration from that.

## Closing

Becoming a contributor to Git has been extremely meaningful to me. It is the first real software that I have actually been a part of, and interacting with people in the Git community has been a great and memorable time. I hope to keep contributing actively, and remain a part of the community as long as I can, maybe forever!

Thanks a lot!

Vinayak