

Improve Google-Drive Support in GVfs



Standard Questionnaire

- What is your e-mail address and IRC nick [Contact Details]?

Student's Name: Mayank Sharma

University Name: Indian Institute of Technology, Kanpur, India

E-Mail Addresses:

1. *****
2. *****

IRC Nickname:

1. mayanksha on GIMPNet (Idle on #nautilus, #gnome-hackers, #libgdata)
2. mayanksha on FreeNode (Idle on #ffmpeg, #ffmpeg-devel and #gsoc)

My Github: <https://github.com/mayanksha>

My GitLab: <https://gitlab.gnome.org/mayanksha>

Other Contact Details:

Skype: <https://join.skype.com/invite/heQnzcMgQOmx>

Facebook: *****

Hangouts: *****

- What is your blog address?

<https://msharma.in>

It just has one post for now, but I have jotted down ideas for some good posts which I plan on writing once my semester is over.

Also, over the 12 weeks duration I'll be writing bi-weekly posts detailing what I achieved.

- **What city and country will you reside in during the summer?**

I plan on staying in Bangalore, India for the majority of my summers. I'll go back to my home-town for once during the summer, but it's just a 3 hour flight, and I'll have internet connectivity (Broadband) at both the places.

- **What is your academic background, including university, major, and expected or achieved graduation date for any degree you have or are pursuing?**

I'm a junior undergraduate pursuing my majors in Computer Science and Engineering from Indian Institute of Technology, Kanpur. I have a keen interest in Systems programming and Software Architecture and have a fairly good knowledge about Networks and *nix systems.

I'm expected to graduate by July, 2020.

- **Who is a possible mentor for the project you are proposing?**

Ondrej Holy is the current maintainer of GVfs and I'd wish he be my mentor. His email address is oholy@redhat.com. He has been very helpful to me, and has introduced me quite well to the GVfs ecosystem (since it has so many different backends).

- **What is the ultimate goal of your proposal?**

Currently, the Google-Backend for GVfs supports only a subset of operations allowed by Google-Drive on the web-interface. The major problem with supporting all the operations is because of the difference in how POSIX systems handle files whereas how a database-backed system like Google-Drive handles files. For a POSIX system, the file path uniquely determines a file (or a link to some other file) and hence file paths are used as the default identifiers. Google-Drive on the other hand identifies files by unique IDs which are generated by Google servers.

This difference results into limitations as to what operations can be performed with the current libgdata APIs. Since, each file's identifier is its ID, i.e. "name" equals ID, we have to specifically use "display-name" for storing a file's title, which is what is shown by nautilus.

Simply copying/moving files from one folder to another results into an error "Operation not supported" so as to preserve the file's title. Copying/Moving is one of the fundamental operations that should be possible on a file. My ultimate goal with this proposal is to make the Google-Drive backend more usable in the following ways:



Primary Goals :

- Support Copy/Move operations on files. Develop the relevant API in libgdata to support updating file objects with `properties`. [\[Issue\]](#)
- Extend the support of deletion to enable deleting shared files. [\[Issue\]](#)
- Write tests for the GVfs Google-Drive backend. Presently, there are no tests and each build on drive backend needs to be manually tested.

Secondary Goals (If time permits):

- Help resolve this Merge Request ([MR !2 \(libgdata\)](#)), since this Project aims to implement document move across folders.
- For the files shared by other users, show some emblem.
- For files having multiple parents, show some emblem to prevent unexpected things from happening. [\[Issue\]](#)

• What applications/libraries of GNOME will the proposed work modify or create?

The plan is to augment libgdata with an API to support storing the ID of the original file in the file update request that is made to the Google-Drive API. As has been documented in the Drive v2 API ([v2 API Reference](#)), the optional `properties` can be used to store the original ID of the file along with any other data which we may deem to be fit. (There's an upper limit as to how much we can store in `properties` and it's pretty low, so we need to think of a good design)

In addition to augmenting libgdata, we'll also have to make changes to GVfs Google-Drive backend and make it work well with the new libgdata API(s). There are also some specific issues pertaining to the Drive Backend (inability to delete shared files) which I'd like to give attention to.

Finally, to test the newly implemented API in libgdata, I will write tests for the same. Apart from libgdata, testing needs to be done with Google-Drive backend as well, and a new test-suite needs to be implemented. Such a test-suite will currently have to be manual, for it needs OAuth Authentication and Authorization to access a Google-Drive but since presently there are no tests at all, we'll be good with these as well. In future, such a test-suite may be added to Gitlab CI but it's fairly complicated, and automating the whole process isn't feasible in a 3 month time-frame.

- **What benefits does your proposed work have for GNOME and its community?**

The issue with copying/moving files in nautilus for Google-Drive backend dates as back as 2015 when it was reported. That time, file copy/move was indeed supported but it resulted into losing display-names for copied files. The relevant bug is on [bugzilla](#).

The resolution of such a high-priority bug implies the following benefits for the GNOME community:

1. Significant improvements to the usability of Google-Drive mounts in nautilus. Add the functionality to copy/move files between folders in nautilus (for Google-Drive mounts).
2. Extension of libgdata with an API which it currently lacks. The `properties` field on file objects can be used to store any kind of key-value pairs. For GVfs, this has immediate benefits since it allows us to store volatile paths.

A number of projects like gnome-photos, evolution-data-server and shotwell list libgdata as one of their dependencies. They may benefit from such a thing in the future if and when they wish to add custom metadata to file objects.

- **Why are you the right person to work on this project?**

Honestly, across the list of ideas, the only project which immediately caught my mind was this. GNOME Shell (and the whole ecosystem) has been my go-to desktop environment since I installed Arch Linux. Nautilus and GVfs have been the most useful apps since I've to frequently access my Virtual Machine allotted by University and an EC2 Instance too for my blog. For file transfers, instead of relying on scp (which is cumbersome), I use sftp in nautilus. Same goes for samba backend as well for accessing files shared by my friends who are on Windows Network.

I have undertaken a rigorous course on Operating Systems wherein we implemented a FUSE based file system and I believe that I learnt a lot from that. Apart from this, I have a good understanding of Networks and Socket API, for I wrote a basic web-server in C to serve static HTML files as well as files of other MIME-types. ([Github](#))

I'm currently doing courses on Software Architecture, and Systems and Network Security wherein we're learning to exploit various types of vulnerabilities and how to patch them. Having an exposure to security always helps while developing libraries and APIs for low-level interaction in C.

From the point of view of language, C/C++, Golang and Typescript have been my go-to choices for whatever projects I have built over the past couple of years. I also have

used the various APIs provided by Google in the past (for various projects) and have a



good knowledge of interacting with them. Having been an avid fan of node.js in the past, I have also learnt about concurrency models and looked into libuv.

As a student of one of the premier institutions in the country, I have learnt how to solve challenging problems and to give my best at everything. Given the chance to do this project under my mentor for GNOME Organisation, I assure to give my hundred percent to it, and would strive to make GVfs more usable.

- **How do you plan to achieve completion of your project?**

My proposed timeline will be as follows:

Dates (2019)	Milestones/Deliverables
Before May, 6	I'll be working on and off with my mentor to resolve bugs here and there.
Community Bonding Period Begins	
May, 6 -- May, 26	With the help from my mentor, I'll be familiarizing myself with the existing codebase and think about a concrete solution to the problem. I'll also work on a couple of bugs from libgdata and GVfs to better transition into this project.
Coding Officially Begins	
May, 27 -- June, 10 (Week 1 & 2)	Write functions to interact with Drive API's file objects. Extend the current `GDataDocumentsService` to include the `properties` field in request. As per Drive v2 API (v2 API Reference), only the update operation supports adding the `properties` in request body.
June, 11 -- June, 24 (Week 3 & 4)	Complete libgdata's API for setting `properties` on file copy/move/uploads. Documenting libgdata's code according to the convention. Write tests for the new API, and thoroughly test it.. Fix any bugs. Create an MR, and put everything into review by maintainers.
First Phase Evaluation	
June, 28 -- July, 10 (Week 5 & 6)	Make changes to design (add/change any data structures) to create room for new libgdata API. First focus on modifying the functions `resolve_dir()` and `g_vfs_backend_google_copy()` and get

	<p>it to work efficiently with volatile paths, while ensuring that changes to <code>`resolve_dir`</code> doesn't break anything else.</p> <p>Test these functions manually for any problems and fix bugs. Create MR and have the Merge Requests in discussion/review by mentors.</p>
<p>July, 11 – July, 22 (Week 7 & 8)</p>	<p>Start working on move operations. Currently, it relies on copy and delete fallback.</p> <p>Add support for move operations. Make changes to the vtable to include move or try_move actions by creating function <code>`g_vfs_backend_google_move()`</code> to support both synchronous as well as asynchronous move operations. Also, make these operations Cancellable.</p> <p>Test the copy/move operations to check it doesn't break anything, and fix any bugs. By the end of this week, the move/copy operations should be working with nautilus.</p>
<h3>Second Phase Evaluation</h3>	
<p>July, 26 – Aug, 9 (Week 9 & 10)</p>	<p>Develop a test-suite for Drive-Backend. There's a lot of edge-cases where Google Drive behaves oddly compared to a normal file system. Most (if not all) of those needs to be implemented and documented for.</p> <p>I plan on writing a man page for <code>gvfsd-google</code> describing the abnormalities of google-drive backend.</p> <p>Refine the GVfs code to make it more presentable. Add final touches to the documentation, and ready the project for submission.</p>
<p>Aug, 10 – August, 26 (Week 11 & 12)</p>	<p>Finalize the Merge Requests after discussion and get the code merged for the next release. Fix other bugs pertaining to GVfs in general.</p> <p>Improve the overall usability of GVfs. If enough time is there, start working on secondary goals as well.</p> <p>During this week, I have planned to attend my first GUADECI, so I'll try to finish most of my work before this week only.</p>
<h3>Final Evaluation and GSoC Results</h3>	



- For GUADEC, the GNOME conference hosted this year 23rd August - 28th August in Thessaloniki, Greece, we encourage you to do a small presentation of 3 minutes. What part of your project will be showable?

Given the opportunity to attend to GUADEC, I'll most certainly attend it and would love to present my approach to the problem at hand in GSoC, and what its solution implies to the uses of Nautilus as a Google-Drive client. I'll also like to present my journey as a first time GSoC'er and how I made my way from a complete newbie to a contributor.

- What are your past experiences with the open source world as a user and as a contributor?

I've interacted with a fair amount of people from the open-source community and so far have had pretty good experiences. From the perspective of a user, I'm a proponent of Open-Source software and I use open-source products wherever possible. In my university, we have fests to introduce people to open-source world and how to begin contributing. For their first non-trivial Pull Request, we also provide students with a T-shirt.

From a contributor's perspective, wherever I've introduced myself as a newcomer, people have welcomed me and have replied to my doubts with full enthusiasm. Apart from GVfs, I've contributed to the cowrie honeypot in the past (1 issue resolved, 1 pending) and to gnome-maps (I have an MR pending there).

For any queries pertaining to GNOME products, I head over to gnome-hackers IRC and people there have been very helpful. All in all, I think very positively of the open source software ideology and that's why I chose to contribute to GNOME organization.

- Please include a link to the committed code in Bugzilla or GitLab merge requests for the issues you fixed for the GNOME module your proposal is related to.

I've so far contributed in the following ways to GVfs project:

- Unification of strings (First Contribution) (Merged)
https://gitlab.gnome.org/GNOME/gvfs/merge_requests/37
- afp: Fix afp backend crash when no username supplied (Merged)
https://gitlab.gnome.org/GNOME/gvfs/merge_requests/38

- Revert "sftp: Always use port 22 if not specified" (Merged)
https://gitlab.gnome.org/GNOME/gvfs/merge_requests/40
- samba: Support "anonymous" user to try anonymous mount first (Pending)
https://gitlab.gnome.org/GNOME/gvfs/merge_requests/41

Apart from these, I'm also working on a patch for this issue:

Support deleting shared Google Drive files (WIP)
<https://gitlab.gnome.org/GNOME/gvfs/issues/265>

- If available, please include links to any other code you wrote for GNOME or other open source projects.

I have worked for the Cowrie Honeygot ([GitHub](#)) and provided an MR

<https://github.com/cowrie/cowrie/pull/1051> (Merged)

I have also worked on the Gnome-Maps application and have an MR

https://gitlab.gnome.org/GNOME/gnome-maps/merge_requests/26/diffs (Pending)

- What other relevant projects have you worked on previously and what knowledge you gained from working on them?

During my Junior year, I took a course on Operating Systems wherein I implemented various syscalls (like sleep, clone, fork, etc) in an x64 based simulator (Gem5). Apart from syscalls, I had also worked on handling page-faults in the kernel space and subsequent page table walks (which I implemented successfully).

Apart from this, I've worked on a FUSE Based file system (BBFS) wherein we were supposed to write the low-level file handling (dirents and inodes) semantics to support a key-value store (keys being files, values being data stored inside files). I opted for a design similar to ext2 having direct, indirect and doubly indirect blocks. As a part of optimization, I also added an inode cache to mitigate slow inode fetches due to disk IO.

I've also worked on a project - "Cryptographically Secure Key-Value Store with Sharing" (written in Golang). The design and code for the same is present [here](#).

During my fresher year, I had finished a project on Linux From Scratch and bootstrapped a SysV Init based barebones distro (from host kernel). I automated the installation of various packages by fetching data from LFS documentation and executing scripts for them. ([Github](#))

- What other time commitments, such as school work, exams, research, another job, planned vacation, etc., will you have between May 6th and August 26th? What are the dates for these commitments and how many hours a week do these commitments take?

I'll be taking GSoC as a full time commitment during my vacations. I have semester holidays starting May, 2, 2019 and I'll be free until July, 25, 2019. I'll be staying at my brother's home in Bangalore and will be utilising most of my productive time towards this project only.

After July, 25, 2019, I'll start with my 7th semester but I have very less load (since it will be my penultimate semester). Hence, I can easily accommodate my schedule to give time to GSoC.

My working times vary a lot (some days I work early in the morning, some days late into the night), yet I can consistently manage around 35-45 hours per week.