

IgnorantGuru's Blog

Linux software, news, and tips

Red Hat / Gnome Developers Censoring GTK+ Bugtracker

I submit the following for your review because it's an interesting case study in how Red Hat developers are running the GTK+ bugtracker, censoring non-flattering input, and misusing their code of conduct. Since they deleted several of my comments, and threatened another participant merely for using the word "overengineered" (lol - if the shoe fits...), I thought it might be valuable to bring what they deleted to larger attention.

I and the others involved are not the only people who are treated this way by these developers. But most people will back down because they don't want to be banned and censored, which I can understand, but it creates an atmosphere where there can be no open discussion of larger issues facing GTK+. But I don't have to kiss ass, and they have never once done anything useful in response to bugs I have tried to get them to fix. In fact I don't think they've ever taken an action that has benefited libre software at all. They are an obstacle - some great upstream to have on your GUI toolkit.

If you plan to use GTK on a new project, don't. Unless you're part of Gnome, this is the kind of support environment you can now expect. And do not be fooled by their "please submit a patch". First, why are they demanding that API users fix their low-level I/O bugs? Second, even the person asking for the patch has no authority to include it - they are more like (ARE) a corporation's customer service representatives that are there to merely give people the runaround.

The case in study is a bug report regarding the way the GTK+ file chooser (file browser) only shows FUSE mounts made by gvfs, and is blind to those made by almost all other file managers. This is a simple bug. All that needs to be done to fix it is add the traditional location used for fuse mounts to the heuristics - a 5 minute job. Yet instead of simply fixing it, first a Red Hat employee immediately closes it as "RESOLVED WONTFIX" with a "No." Then after I point out some details, they reopen it, but embroil it in a huge debate about gvfs dependency and udisks, which has nothing to do with this simple bug. As such, they are obstructing, not resolving anything. When their inaccurate gvfs dependency information is pointed out, they delete the comment. Further, it's revealed that it's broken in the first place because someone inserted a hack for gvfs into gio, breaking the chooser for non-gvfs use. When this is pointed out, they delete the comment.

You should recognize some of the names involved from previous articles on this blog. Emmanuele Bassi states he doesn't work for Red Hat. I don't know who he does work for, but he is very prominent in Red Hat's projects (allegedly open and free projects, but in which Red Hat dictates all decisions), and he seems to be given carte blanche by them administratively, for whatever reason.

Here is the full thread, with comments deleted by Red Tape restored and indicated.

GNOME Bugzilla - Bug 750182 (https://bugzilla.gnome.org/show_bug.cgi?id=750182)

GtkFileChooser should also search for mountpoints in \$HOME/.cache

Last modified: 2015-06-04 13:21:15 UTC

Summary: GtkFileChooser should also search for mountpoints in \$HOME/.cache

Status: REOPENED

Product: glib

Component: gio

Assigned To: gtkdev

Description Psy[H] 2015-05-31 19:30:12 UTC

\$HOME/.cache is a logical place for applications to place user-level mountpoints such as FUSE filesystems. Those should be visible in GtkFileChooser. Right now it seems to exclude all hidden dirs in \$HOME from mountpoint search. \$HOME/.cache should be whitelisted.

Comment 1 Matthias Clasen 2015-06-01 16:06:23 UTC

No. XDG_CACHE_DIR has defined semantics that don't match mounting things.

[Marked RESOLVED WONTFIX]

Comment 2 IgnorantGuru 2015-06-02 15:18:22 UTC

I realize that GTK development is effectively dead for non-Gnome projects, but you may wish to consider that several widely used XFCE and LXDE file managers which use GTK, such as Thunar and PCManFM, as well as other programs, do mount fuse and network filesystems in XDG_CACHE_DIR, and have done so for years. This is common usage, and not every GTK project follows Freedesktop specs to the letter (specs which are poorly maintained and often incorrectly documented and implemented). fuse filesystems are mounted in a user-writable directory, usually somewhere \$HOME, and it's poor form to create non-hidden directories in the user's home. The file chooser as it stands apparently ignores all hidden directories, which doesn't leave good options.

Since the GTK File Chooser does discovery of mounted volumes, and since the .cache location has been commonly used for this for years, it would be helpful to the general set of software which uses GTK if the chooser listed volumes mounted in these common locations as well as those used by Gnome/Freedesktop. Otherwise it is blind to them and fairly useless outside of Gnome.

Just quoting the specs and ignoring common usage to avoid updating it just makes the file chooser irrelevant for real uses. Is GTK now documented as a Freedesktop/Gnome-only project? You have several non-Gnome file managers using GTK, so perhaps supporting their uses as well as Gnome would be appropriate.

Comment 3 Matthias Clasen 2015-06-02 15:41:20 UTC

> Right now it seems to exclude all hidden dirs in \$HOME from mountpoint search.

> \$HOME/.cache should be whitelisted.

The file chooser does not do any mountpoint search at all. We rely on gvfs to provide this information.

> GTK development is effectively dead for non-Gnome projects

GTK development for non-gnome projects depends on developers from those projects participating and making their needs heard.

> The file chooser as it stands apparently ignores all hidden directories, which > doesn't leave good options.

That is not true.

Comment 4 IgnorantGuru 2015-06-02 15:48:43 UTC

> The file chooser does not do any mountpoint search at all. We rely on gvfs to provide this information.

Then how is it that the GTK file chooser still finds volumes even when gvfs is not installed? (In case you didn't know, GTK can be and is used without gvfs.)

> GTK development for non-gnome projects depends on developers from those projects participating and making their needs heard.

We are doing so here, and as usual, those needs are summarily dismissed if not coming from Red Hat.

> That is not true.

Perhaps you can clarify, since the file chooser documentation says nothing.

Comment 5 Psy[H] 2015-06-02 16:18:25 UTC

> The file chooser does not do any mountpoint search at all. We rely on gvfs to provide this information.

I can confirm that I do not have any gvfs package in my system, but GtkFileChooser still successfully finds mountpoints in /media, /run/media/\$USER, \$HOME/ (excluding hidden dirs).

> No. XDG_CACHE_DIR has defined semantics that don't match mounting things.

Is there any standardized place for user-level mounts that won't noticeably interfere with user's home dir but GtkFileChooser would find?

Comment 6 Matthias Clasen 2015-06-02 16:39:14 UTC

(In reply to Psy[H] from comment #5)

> > The file chooser does not do any mountpoint search at all. We rely on gvfs to provide this information.

>

> I can confirm that I do not have any gvfs package in my system, but

> GtkFileChooser still successfully finds mountpoints in /media,

> /run/media/\$USER, \$HOME/ (excluding hidden dirs).

>

> > No. XDG_CACHE_DIR has defined semantics that don't match mounting things.

>

> Is there any standardized place for user-level mounts that won't noticeably

> interfere with user's home dir but GtkFileChooser would find?

I'll move this bug to glib - I'm not 100% sure what heuristics `gunixmounts.c` applies when looking for mounts.

Comment 7 Emmanuele Bassi (:ebassi) 2015-06-02 16:54:53 UTC

(In reply to Psy[H] from comment #5)

> > The file chooser does not do any mountpoint search at all. We rely on gvfs to provide this information.

>

> I can confirm that I do not have any gvfs package in my system, but

> GtkFileChooser still successfully finds mountpoints in /media,

> /run/media/\$USER, \$HOME/ (excluding hidden dirs).

>

> > No. XDG_CACHE_DIR has defined semantics that don't match mounting things.

>

> Is there any standardized place for user-level mounts that won't noticeably

> interfere with user's home dir but GtkFileChooser would find?

Non-system wide FUSE mount points should really go in the user's XDG_RUNTIME_DIR; using XDG_CACHE_DIR is a known fallback used in the past when XDG_RUNTIME_DIR did not exist - but should really be ignored (e.g. GVFS will fall back to \$HOME/gvfs instead if it finds out that XDG_RUNTIME_DIR is XDG_CACHE_HOME, since it's where GVFS mounts FUSE file systems).

The strong suggestion is for file managers willing to interoperate with the system and various toolkits to follow the basedir specification; if the spec is unclear, asking on `xdg-list` yields timely replies.

(In reply to Matthias Clasen from comment #6)

> I'll move this bug to glib - I'm not 100% sure what heuristics `gunixmounts.c`

> applies when looking for mounts.

`g_unix_mount_guess_should_display()` will discard system directories (like `/proc` or `/sys`), but only uses user-accessible mount points under `/media` and `/run/media/$USER`. The check for `'/run'` is hard-coded, which is not right: it should get the `XDG_RUNTIME_DIR` environment variable instead.

Comment 8 Psy[H] 2015-06-02 17:06:52 UTC

Thanks for comments!

GtkFileChooser does not find mount in `$XDG_RUNTIME_DIR/mountpoint` (3.14.5-1).

Comment 9 Emmanuele Bassi (:ebassi) 2015-06-02 17:24:45 UTC

(In reply to Psy[H] from comment #8)

> Thanks for comments!

> GtkFileChooser does not find mount in `$XDG_RUNTIME_DIR/mountpoint` (3.14.5-1).

It won't, as I said in comment #7.

If no GVFS is present, the fallback code will list the Unix mounts coming from `/proc/mounts` or `/etc/mtab`; each mount point will be checked via `g_unix_mount_guess_should_display()`, which will return `TRUE` for user-accessible mount points under `/media`, `/run/media/$USER`, or directly under `$HOME`.

If you want your mount points to be visible in the GTK file chooser and you don't have GVFS running, you should mount them under `"/run/media/$USER"`.

The bug in GIO is that the code doing the check hardcodes `/run` instead of using `XDG_RUNTIME_DIR`. If `XDG_RUNTIME_DIR` is unset, it should fallback to `XDG_CACHE_DIR` (which is what `g_get_user_runtime_dir()` does), but that would mean that the FUSE mount points would go under `XDG_CACHE_DIR/media/$USER`, which is a bit ridiculous.

Hence my suggestion is to set `XDG_RUNTIME_DIR` to `/run` and create the `/run` directory at system start up, and remove it at system shutdown, following the basedir specification recommendation.

Comment 10 Psy[H] 2015-06-02 17:31:36 UTC

`/run/media/` AFAIK is usually created by `udisks`, and here are two difficulties:

- `udisks` moved mounts to `/media/$USER/`

- `udisks` is not installed on every system.

Since `/run` is not user-writable, there could be no tool to create `/run/media/$USER` in the system.

Comment 11 Psy[H] 2015-06-02 17:35:21 UTC

According to spec, `$XDG_RUNTIME_DIR` should be owned by user, it can not be set to `/run`.

Comment 12 Psy[H] 2015-06-02 17:38:37 UTC

As far as I can tell, `$XDG_RUNTIME_DIR` is typically resolved into `/run/user/$UID`

700 permissions are mandatory by the spec.

Comment 13 Psy[H] 2015-06-02 17:51:07 UTC

`/run/user/$UID` is a `tmpfs` mount by itself. Maybe that is why `GtkFileChooser` does not see other mounts inside, despite it looks into `/run`. If that is so, then search logic should be changed to search through `tmpfs` mounts for user's mounts in case of `$XDG_RUNTIME_DIR`

Comment 14 IgnorantGuru 2015-06-02 18:01:56 UTC

Based on traditional use, `gio` should look in `XDG_CACHE_HOME` for mounts, regardless of the current specs. For example, file managers traditionally mount fuse to `~/cache/program-name/mount-point`. This makes some sense since `XDG_RUNTIME_DIR` used to (or does?) fall back to a cache dir.

The point here is not to design a spec from scratch (yet again), but to realize that there are already many GTK apps using that location for mounts (with users trying to find mounts in that location), based on traditional use. If you merely support a new spec then the file chooser still won't show mounts created by most file managers in use. In other words, it won't have practical value, no matter how spec-compliant. It seems like the only exception you support is `gvfs's ~/gvfs`, ignoring traditional use of fuse.

Arguing what's theoretically best while ignoring what's already in use and established is not of any practical value. This means GTK's file chooser will continue to be blind and useless for finding mounts, unless it searches `XDG_CACHE_HOME` to a reasonable depth.

Comment 15 IgnorantGuru 2015-06-02 18:08:19 UTC

Also, `"XDG_CACHE_DIR"` isn't even in the spec, you apparently mean `XDG_CACHE_HOME`, so it's not helping clarity by referring to non-existent variables. eg <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=736448> (<https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=736448>)

Comment 16 OmegaPhil 2015-06-02 18:52:06 UTC

I would also appreciate `XDG_CACHE_HOME` being consulted for mounted filesystems.

Comment 17 IgnorantGuru 2015-06-02 21:26:50 UTC

> Non-system wide FUSE mount points should really go in the user's `XDG_RUNTIME_DIR`; using `XDG_CACHE_DIR` is a known fallback used in the past when `XDG_RUNTIME_DIR` did not exist - but should really be ignored (e.g. GVFS will

fall back to \$HOME/.gvfs instead if it finds out that XDG_RUNTIME_DIR is XDG_CACHE_HOME, since it's where GVFS mounts FUSE file systems).

Why should it "really be ignored", when it "is a known fallback used in the past" (iow it is still widely used)? That is the real source of this bug - that gio included a gvfs-specific hack, breaking Freedesktop. If g_get_user_runtime_dir() falls back to XDG_CACHE_HOME, then that is what should be searched. Nowhere in ~/.freedeskop does it mention "~/.gvfs", and I doubt glib suggests ignoring g_get_user_runtime_dir().

Most glib apps will not read XDG_RUNTIME_DIR directly, they will use g_get_user_runtime_dir() - that's what it's for. It does indeed fallback to "~/.cache" on Debian at least. In fact this is probably why mounts have been placed there for years by many file managers, and why the chooser is blind to those mounts - it's not looking in XDG_RUNTIME_DIR or its glib fallback and predecessor, XDG_CACHE_HOME.

> but that would mean that the FUSE mount points would go under XDG_CACHE_DIR/media/\$USER, which is a bit ridiculous.

No, it would generally mean mounts are placed beneath XDG_CACHE_HOME, such as XDG_CACHE_DIR/Thunar/mount-point (which is exactly the case). Apps can use the cache as they please. Discovery should be reasonably general, not specific to just gvfs behavior and hacks.

So the question is, why does gio switch to a gvfs hack instead of searching for mount points in g_get_user_runtime_dir() ? If you want to include gvfs hacks in gio's discovery, so be it, but you should not break existing apps that are using g_get_user_runtime_dir() without such hacks.

So I think the proper behavior is to search g_get_user_runtime_dir(), at least down to a few subdirs. That will fix this bug and also preserve discovery of common mount points (which will continue to be created in ~/.cache for years in practice, even if you now change the spec to match gvfs hacks). For even better discovery, it should look expressly in ~/.cache in addition to XDG_RUNTIME_DIR/XDG_CACHE_HOME, since that location is commonly used. The whole point of mount point discovery is convenience, not contrived spec compliance (and hacks) to the point where it only works with a small subset of systems. That's why the file chooser is blind - you're basing it on theory rather than actual practices, which isn't much good for discovery.

Comment 18 IgnorantGuru 2015-06-02 21:44:22 UTC

Also, you may want to consider that the reason /run was hardcoded is that XDG_RUNTIME_DIR may rarely be set. It isn't in Debian, and g_get_user_runtime_dir() returns "/home/user/.cache".

Also, if XDG_RUNTIME_DIR defaults to /run/media/\$USER on some systems, that may not be appropriate for fuse mounts. And /run does not allow the user to write to it, again no good for fuse mounts. XDG_RUNTIME_DIR is also sometimes set to /tmp, which may have noexec and other restrictions. These are reasons why file managers may use XDG_CACHE_HOME as a more reasonable default, and why the chooser should search all locations. Personally, I've never seen such mounts in /run, except by root daemons such as the old udisks2 behavior (which was eventually deemed non-FHS compliant and moved elsewhere).

All of that said, for practical discovery, it would be smart to search /run (hardcoded), XDG_RUNTIME_DIR, XDG_CACHE_HOME, AND \$HOME/.cache, if they differ.

Comment 19 Emmanuele Bassi (:ebassi) 2015-06-02 21:45:36 UTC

(In reply to OmegaPhil from comment #16)
> I would also appreciate XDG_CACHE_HOME being consulted for mounted > filesystems.

I would gladly review a patch that added that to the checks inside g_unix_mount_guess_should_display(). I'm not a GIO maintainer, though, so you will need somebody else's ACK for it.

(In reply to Psy[H] from comment #10)
> /run/media/ AFAIK is usually created by udisks, and here are two > difficulties:
> - udisks moved mounts to /media/\$USER/
> - udisks is not installed on every system.

To be absolutely blunt, not installing components and then complaining that things are broken is not really cool. It's not like we want to duplicate the logic everywhere: we put it inside some component for a *reason*. GIO depends on GVFS on Linux, and GVFS depends on udisks. If you're using some other OS, the chain of dependencies is different, but we kind of treat the stack as a stack, not as a pick and mix bowl of "may be nice to have". The reason the dependency is "soft" (i.e. we don't make GIO depend some libraries) is mostly a case of 1. historical accidents; 2. a convenience for integrators to avoid dependency cycles; and 3. because on Windows, *BSD, or MacOS, the dependencies are fairly different.

In any case, there's nothing that says that udisks *must* be the component creating the /run/media/\$USER directory.

Finally, /media is also another location that is checked when going through the list of mount points.

> Since /run is not user-writable, there could be no tool to create > /run/media/\$USER in the system.

Anything that creates /run can also create /run/media/\$USER when the user session starts; since it's going to be a privileged user, it can change the directory's permissions as well.

Anyway, I'll stand by what I wrote at the top: I'll gladly review a patch that adds a check for a user-accessible mount point under XDG_CACHE_HOME.

Comment 20 IgnorantGuru

COMMENT DELETED by André Klapper

> To be absolutely blunt, not installing components and then > complaining that things are broken is not really cool. It's not like > we want to duplicate the logic everywhere: we put it inside some > component for a *reason*. GIO depends on GVFS on Linux, and GVFS > depends on udisks.

Actually, you're being absolutely inaccurate. gio does not depend on gvfs - it is part of glib and runs fine without gvfs. Perhaps you should review what a dependency is. From your own docs: "One of the big advantages of putting the VFS in the GLib layer is that GTK+ can DIRECTLY use it, e.g. in the filechooser." <https://developer.gnome.org/gio/stable/ch01.html> (<https://developer.gnome.org/gio/stable/ch01.html>)

low, GTK+ does NOT depend on gvfs for a reason. The point of putting it in the glib layer was to avoid GTK+ dependencies on DE-specific filesystem abstraction layers like gvfs. I really [sic - realize] Red Hat has done everything in their power to break that separation and create a monolithic stack, but for now glib is not gvfs dependent.

> Finally, /media is also another location that is checked when going > through the list of mount points.

That's not useful for fuse, since the user cannot write to /media, and not all systems use acls.

> Anyway, I'll stand by what I wrote at the top: I'll gladly review a > patch that adds a check for a user-accessible mount point under > XDG_CACHE_HOME.

So no gio maintainers are willing to maintain gio to correct gvfs hacks they included that break Freedesktop, even though they're the ones who broke it in the first place. Thanks for being predictable.

Perhaps you can point us to the gio discovery function at least? Or is that too much trouble too? But I doubt I would waste time on coding as I'm sure you'll just make an excuse to reject it, just as you're making excuses instead of fixing this bug.

Comment 21 IgnorantGuru

COMMENT DELETED by André Klapper

For people who don't know the history on this, modern GTK devs (iow Red Hat - some of the same names we see here) tried desperately to make GTK dependent on gvfs. However, it broke everyone's work and gvfs didn't work everywhere, so they were forced to backtrack. To no one's surprise but theirs, GTK still runs just fine without gvfs (except where they deliberately break it, like this gvfs hack interfering with other software), but I'm sure it's still their agenda to make it a dependency just because they want it to be, and the inaccurate information being given here is right in line with that history.

So basically gio is only now supported for gvfs use at most - they won't even think of making or accepting any changes which help other general users of GTK. And I've heard this "please submit patches, test cases, more info", etc. before from these same people. That is their way of just ignoring you - they'd rather waste your time than tell you flat out that they refuse to support gio (or really GTK), and won't accept any changes that do so. At least that has been the pattern of behavior.

Comment 22 Psy[H] 2015-06-03 08:25:21 UTC

> To be absolutely blunt, not installing components and then complaining that things are broken is not really cool.

I also disagree, GTK has nothing to do with udisks. Only gvfs-daemons has dependency on udisks, and we are not touching that here. Plus, udisks does not use /run/media/\$USER anymore anyway. And /media/* is not user-owned, so it's no good for fuse.

\$XDG_RUNTIME_DIR also seem to be only maintained by systemd-related components which is not universal. So, fallback to \$XDG_CACHE_HOME or \$HOME/.cache seems reasonable. There are no non-homedir user-owned locations that do not depend on optional overengineered stuff like udisks, gvfs, systemd, etc. \$XDG_CACHE_HOME or \$HOME/.cache is the only 100% backwards compatible fallback.

Comment 23 Emmanuele Bassi (:ebassi) 2015-06-03 08:54:58 UTC

(In reply to IgnorantGuru from comment #21)
> For people who don't know the history on this, modern GTK devs (iow Red Hat > - some of the same names we see here)

This will be the only time I reply to one of your comments, and it's also your final warning. Bugzilla is under the code of conduct of GNOME, and you have been consistently rude, dismissive, and flat out insulting in every single interaction with the developers.

Either you stop, or you get your account revoked.

Comment 24 Emmanuele Bassi (:ebassi) 2015-06-03 09:06:39 UTC

(In reply to Psy[H] from comment #22)
> > To be absolutely blunt, not installing components and then complaining that things are broken is not really cool.
>
> I also disagree, GTK has nothing to do with udisks.

As you may have noticed, it does.
> Only gvfs-daemons
> has dependency on udisks, and we are not touching that here.

GIO has a soft-dependency on GVFS; the reason why GIO does not implement the functionality of GVFS directly inside its code base is one of expedience and historical reasons; GVFS had to be implemented as a separate code base to avoid adding dependencies to GLib. Nevertheless, on Linux, GIO is pretty much dependent on the functionality provided by GVFS, and it falls back to internal implementations, but the fall backs are not heavily tested.
> Plus, udisks does not use /run/media/\$USER anymore anyway.

Nevertheless, /run/media/\$USER is what modern Linux uses for user-accessible mount points - including FUSE.
> And /media/* is not user-owned, so it's no good for fuse.

At no point I've said that FUSE mount points should go under /media; I've just listed /media as another location used.
> \$XDG_RUNTIME_DIR also seem to be only maintained by systemd-related > components which is not universal.

It should be universal, and it's not systemd-related; that's why it's in the basedir specification on fd.o.

XDG_RUNTIME_DIR was introduced because XDG_CACHE_HOME is not a good place for storing session-related files in a secure way. Please, read the discussion that led to the creation of XDG_RUNTIME_DIR:

<http://lists.freedesktop.org/archives/xdg-list/2010-November/011681.html> (<http://lists.freedesktop.org/archives/xdg-list/2010-November/011681.html>)

> So, fallback to \$XDG_CACHE_HOME or \$HOME/.cache seems reasonable.

As I said (and I won't say it again), I'd gladly review a patch that adds this; you'll have to also convince a GLib/GIO maintainer. You should join the #gtk+ channel on irc.gnome.org.
> There are no non-homedir user-owned locations that do not depend on optional > overengineered stuff like udisks, gvfs, systemd, etc.

Please, refrain from making comments like these in the future. The reason why things like GVFS or udisks are complex systems is because they solve real problems. If you decide to not use them because your requirements are simple or unchanging it does not invalidate the problems and requirements of people actually using them.

Comment 25 André Klapper 2015-06-03 12:39:34 UTC

Psy[H]: See comment 23 and read <https://wiki.gnome.org/CodeOfConduct> (<https://wiki.gnome.org/CodeOfConduct>) . Thanks.

Comment 26 Psy[H] 2015-06-03 15:05:19 UTC

GtkFileChooser can successfully look for mounts without udisks and gvfs installed. The sole purpose of this bug report is to tweak existing behavior. At least to look into \$XDG_RUNTIME_DIR according to basedir spec. just like Emmanuele Bassi said:

> g_unix_mount_guess_should_display() will discard system directories (like /proc or /sys), but only uses user-accessible mount points under /media and /run/media/\$USER. The check for '/run' is hard-coded, which is not right: it should get the XDG_RUNTIME_DIR environment variable instead.

As for \$XDG_CACHE_HOME, I've asked about proper XDG_RUNTIME_DIR fallbacks on XDG mailing list, waiting for reply.

Comment 27 OmegaPhil 2015-06-03 19:33:12 UTC

Responding as I was quoted here - sorry but I'm not the maintainer, I have my own projects I'm struggling with (feeling the pain of C++...) so I won't be working on a patch (ha, at least currently with my progression I doubt I could make such a patch and have it pass muster). The problem is simply occasionally annoying for me, its not a big deal.

I see IG's post has been deleted, censorship is unacceptable (for the record I have no interest in a 'code of conduct', if you see shit you call it out, although currently I don't have any personal grudge here). This is the correct place to discuss and hash out problems and presumably should act as the public record, so I've summarised IG's points here without his anger at least (to preempt things, just because you don't like this doesn't mean the information is not useful for others coming to this ticket):

- o GTK is not supposed to be dependent at all on GVFS.
- o In recent times, Red Hat developers have attempted to make it dependent (e.g. the 'soft dependency' mentioned previously).
- o This broke a lot of things, thus a more serious dependency had to be backtracked.
- o With the current situation and this bug, it looks like the maintainers don't care to be compatible/accessible with client software that already has well-established behaviour, and is not involved with the GNOME software stack.

Basically hes very concerned that with this and other issues, important core software is being stolen away from normal non-GNOME usage.

Comment 28 Emmanuele Bassi (:ebassi) 2015-06-03 20:03:07 UTC

(In reply to OmegaPhil from comment #27)
> I see IG's post has been deleted

Nothing has been deleted; he removed himself from the Cc: of this bug. Stop spreading false accusations.

> censorship is unacceptable (for the record I have no interest in a 'code of conduct',

It does not matter if you have no interest: the code of conduct for GNOME services exists, and it's enforced on Bugzilla.

if you see shit you call it out,

No, it does not work that way.

You behave like a decent human being, and you afford some level of courtesy to the people that work on the stack that you use. You assume people mean well, and you treat them like intelligent people that are trying to solve problems and work on an open, volunteer-driven project.

> o GTK is not supposed to be dependent at all on GVFS.

While GTK does not depend on GVFS, GTK depends on a set of functionality that is *implemented* by GVFS. If nothing implements it, then GTK simply will fall back to a subset of that functionality.

> o In recent times, Red Hat developers have attempted to make it dependent > (e.g. the ‘soft dependency’ mentioned previously).

Red Hat does not enter in the picture; I’m not a Red Hat employee. You can leave you conspiracy theories at the door.

> o This broke a lot of things, thus a more serious dependency had to be > backtracked.

This has broken nothing. Not showing FUSE mount points from random, unspecified locations that just so happen to be used by some project is not a break in functionality.

> o With the current situation and this bug, it looks like the maintainers > don’t care to be compatible/accessible with client software that already has > well-established behaviour, and is not involved with the GNOME software > stack.

I’ve said *multiple* times that I’ll gladly review a patch; I also said that, while I routinely work on the G* core platform, I’m also not a GIO maintainer, thus I cannot guarantee that the patch I review will be integrated. You can try and convince a GIO maintainer — join IRC and state your case.

What I’m not going to do is write the patch for you, since I don’t have any issue with the current stack, and I also have other projects — as well as work — that I maintain and have to care about.

Since the people in this bug use applications that do not conform to the basedir specification it’s entirely up to them to pursue a fix in GIO.

> Basically hes very concerned that with this and other issues, important core > software is being stolen away from normal non-GNOME usage.

Honestly, if 1% of the energy spent finger-pointing, complaining, and resorting to rude and unfounded accusations were instead spent writing the patch in question, we could have closed this bug already.

Comment 29 OmegaPhil 2015-06-03 20:09:42 UTC

Take a look at the page here: https://bugzilla.gnome.org/show_bug.cgi?id=750182 (https://bugzilla.gnome.org/show_bug.cgi?id=750182)

You quoted his post that has been removed – I searched the page for ‘history’ and found his post had gone – thats all.

Literally, I don’t have strong feelings for this, I have just posted his points and you have responded, which is fine.

Comment 30 André Klapper 2015-06-04 13:20:21 UTC

[offtopic]

(In reply to OmegaPhil from comment #27)

> I see IG’s post has been deleted, censorship is unacceptable

Your interpretation of “censorship” is incorrect. See <http://xkcd.com/1357/> (<http://xkcd.com/1357/>)

In the GNOME community, the GNOME Code of Conduct applies. (And technically, I have hidden two postings from being shown to non-admins.) [He says this after they stated previously that nothing was deleted – always nice to be technically accurate while you’re lying.]

GNOME Bugzilla is indeed the correct place to discuss and hash out problems and your input is very welcome IF you stick to technical aspects instead of personal attacks.

Editorial: Personally, I don’t see discussing Red Hat’s motivations and history of involvement in this area as irrelevant or as a personal attack. There was no name calling here by anyone, really no references to any person in particular. In my view they are using their code of conduct (which also specifies that we must assume they mean well!) merely to hide misinformation they’re spreading about gvfs, and to hide the involvement of Red Hat and their history and agendas with regard to gvfs dependency. When a person uses the word “overengineered” to justify why they’re not using udisks (which they shouldn’t have to justify at all, as it’s a valid choice and nothing to do with this bug), and they are threatened with a code of conduct for using the word, there is something seriously wrong. So effectively, none of these central issues can be discussed on a GTK+/glib bugtracker without heavy-handed censorship and threats.

I feel personally attacked because Emmanuele Bassi called us “not really cool” for thinking we can pick and choose components in Linux.

June 4, 2015 - Posted by [IgnorantGuru](#) | [Uncategorized](#)

11 Comments [»](#)

1. Judging solely from this transcript, methinks that Messrs. Klapper and Bessi have an inverted notion of who’s being rude and dismissive, and who has an informed and relevant opinion.

Comment by IA | June 4, 2015 | [Reply](#)

2. I also submitted an email complaint to the gtk-list and gtk-app-devel-list. These will be mostly fanboys responding, since that’s all that’s left in the so-called GTK+ community (only those who have not been completely disregarded and driven out by recent GTK 3 development practices remain, so it’s a dismal pool for peer support), but I felt calling them on this behavior is important, at least on principle. We’ll see how the ‘community’ handles these abusive admin complaints. Should they delete this archive, let me know and I will republish the emails.

<https://mail.gnome.org/archives/gtk-list/2015-June/thread.html>
<https://mail.gnome.org/archives/gtk-app-devel-list/2015-June/thread.html>

Comment by IgnorantGuru | June 4, 2015 | [Reply](#)

3. Bassi is a long time dev for GNOME and Intel, he is the author of Clutter, among other things. So if technically he may have or not have a title in the project, he may be granted to boss around.

Still asking myself why are you people still using that entitlement development kit.

Comment by Theodore | June 4, 2015 | [Reply](#)

o > Still asking myself why are you people still using that entitlement development kit.

A fair question. I think most independent devs still using GTK are using it because they originally started using it when it was a free and open project (the whole Gnome universe used to be a community-driven response to such idiocy, but has since been consumed by the same corporations it once opposed). They continue to use it because like me, they have thousands of hours of work invested in their projects, and want to get a decent lifespan out of them.

In my case, I forked an early PCManFM, before I really knew what was going on in GTK and glib dev. To just dump it now would be a major rewrite and redesign, which is a whole fucking lot of work, to put it simply, especially for a few developers on a small project. It’s less work to put up with a few things for now, but obviously if they get out of hand that’s no longer true. That’s also why I kept GTK2 support intact, and removed many RH stack dependencies (like udisks, libgudev, etc). SpaceFM can survive on glib, GTK2, and eudev if needed. I did not jump fully on the GTK3 boat, and I think those that do will regret it, especially if they expect any freedom from systemd, gvfs, and god knows what other nightmares they have in store.

As for this bug, it doesn’t directly affect SpaceFM much – I just gave my input to point out what the issue really was. It does affect all users of independent file managers and apps somewhat. It’s a minor bug, which makes their response all the more ridiculous and out of proportion. But it’s a classic example of their mismanagement of what was once an open and community-developed project. They’re trashing it rather than forking it, overwriting and breaking (as in this bug) the work of many people who contributed.

I think if any independent Linux community remains, it should fork GTK2 as GTK4, and fork glib as well, since they will likely poison it further. Nor is it GTK alone – much of this applies to the whole of GNU Linux. Fork it before it’s too late, because they’re taking it from you piece by piece.

Comment by IgnorantGuru | June 4, 2015 | [Reply](#)

o Also, as has already been discussed on this blog, the only other GUI toolkit that can perform comparably is Qt, and that has its own corporate agendas, is corporate controlled, and used the same Red Hat stack. Gnome was the DE developed independently, sort of like XFCE is today, but it was basically hijacked by big money interests. Unfortunately, XFCE and LXDE devs are becoming increasingly dependent on the Gnome stack, using conveniences (traps) like gvfs, udisks, systemd, etc. It’s not easy to escape from once you invest in their corporate-controlled tools.

To me, it’s amazing that the Linux community can’t fork or create a decent GUI toolkit after all this time. As a result, they give all control of their OS away. (MATE, Devuan and others are making an honest effort though.)

Comment by IgnorantGuru | June 4, 2015 | [Reply](#)

o In my opinion there is EFL, which is driven by Samsung and has its agenda... And there is TCL/TK. Both ways require a rewrite and trashing some parts of the logic behind, so the only alternative to GTK is GTK... Only at starting to think at forking a backport of that mess I get migraines, but it’s the way to go after all. :)
Imho there are big parts of the foss community who “are ok with”, like I’m ok with preparing the moksha desktop on bodhi linux for my relatives and use cwm with some scripts, instead of changing pc to use gnome or kde. See Linus Torvalds and his app for diving, ported to qt and everything is ok.

And don’t forget the fact that, yeah, St. Ignucius looks at us silly users of Linux, but his operating system has nearly everything needed to be free (and when gnumeric and abiword will fuck up he will simply ask to rewrite them in gnustep, for example), the kernel bared and works too. Recently they asked him about systemd. He has dmd and answered that [systemd] it’s free software, so anybody can judge accordingly.

I have some shit I strive (against my laziness) to do, can we at least start a discussion about which parts of freedesktop are to be forked, and where (in its history) gtk should be forked?

Comment by Theodore | June 4, 2015 | [Reply](#)

o Funny that you mention RMS... as respectable as his views are in general, I think his political tendency to embrace the assholery that’s directed at anti-imperialistic nations opens the door to effectively imperialistic incursions against free software, which is the problem we’re seeing here.

Comment by IA | June 4, 2015 | [Reply](#)

o I despise and dislike a lot of things about Stallman and the GNU, from the horrific blend of lisp ideas and unix parts to the turboliberalism the man promotes. But I have to admit the GNU project is a workforce that doesn’t act for mere profit (period), doesn’t give a fuck about hurt corporate feelings, and has no fear of forking big ass codebases without rewriting the whole mess. Probably it’s unique in its genre nowadays: the communities who are the most similar to it are backed or care for diplomacy. But it enters in active duty only to fight corporate freedom threats. And will not do anything for the shit gtk is turning into, ‘cause there is no freedom threat into worsening a software under gpl. And, whether their software using gtk will experience serious issues, they will simply ship an emacs plugin and a gui in gnustep.

Comment by Theodore | June 5, 2015 | [Reply](#)

o I mostly view Stallman as a fraud – and I’ve had a few email conversations with him that certainly did not dispell this – he is a devoted enabler of Red Hat corp, and other corporations. ‘Control the opposition’ is hardly a new concept, and I think the FSF is how corporations ensure that free software doesn’t evolve. Mostly they have seen to that. Stallman is paid well to travel around and pretend.

TCL/Tk has no internationalization, and lacks other features. GTK and Qt are pretty much the only reasonable choices for an app like SpaceFM.

I would say GTK should be forked either from 2, or from before they started removing eg GtkImageMenuItem and changing default behavior for mnemonics, etc. 3 has its advantages in rendering, etc – if it was developed with a stable theming API, etc., it could still be decent. To me now it looks ugly, and is too always-broken to use – I have moved back to GTK2.

But GTK and even systemd are not the only things – Red Hat has a stack from kernel patches, kdbus pushing into the kernel, systemd, udev, udisks, gvfs, Xorg, gtk. That whole stack is poisonous and loaded with security problems and horrible engineering. That’s why I don’t have much hope for Linux. systemd alone turns it into a Windows piece of garbage.

I’ve liked some of what I’ve read about OpenBSD (they seem more against systemd than FreeBSD, and I like their simplicity approach to security and defaults), so longer term I’m thinking of slowly moving into that, and maybe porting SpaceFM to it, with some native device handling for BSD. Since SpaceFM is mostly just gtk and glib at this point, it should be doable (maybe replace udev with some sysfs polling, or however it’s done in BSD). And the Linux build should continue to work as well.

However, it should be noted that BSD on the desktop still uses Red Hat’s Xorg, so it can hardly be considered secure. And it uses Gnome as its typical DE. I plan to install just X and Openbox and try that. I view it as a better long-term investment than distros trying to avoid systemd and the rest of RH’s stack, as noble as their efforts are.

Comment by IgnorantGuru | June 7, 2015 | [Reply](#)

o I don’t follow people enough to know who Bessi is. Maybe the conversation doesn’t cast him in a good light, but it doesn’t feel right to give much weight to a long track-record if the person doesn’t believe in software independence at lower levels.

Comment by IA | June 4, 2015 | [Reply](#)

o Emmanuele Bassi, Klapper, and Clasen are not new to this blog. They were quoted in some of the GTK articles I’ve written, and always sound like arrogant asses (I think everyone takes classes in that at Red Hat). They received a lot of flack from those articles. I think they’re like customer service trolls hired by Red Hat to run and monitor the Gnome-related bugtrackers, etc. They mostly obstruct any genuine community involvement and any non-RH-approved changes, and push RH’s monolithic stack as the only possible Linux. Just like they did here.

For my part, it’s not a question of whether I personally want to use systemd or gvfs. But as an independent FM dev, I want my software to be able to run on a wide variety of systems, not just Gnome. SpaceFM has only gtk, glib, and udev as core dependencies, allowing it to run almost anywhere, and this took work to achieve. If they connect gtk to gvfs with a hard dependency, then my users have to install all of Gnome just to use it, similar to how you have to install KDE to use krusader. That’s how corporations think they achieve dominance. But that just causes devs to drop GTK.

If they were smart, they’d realize that making a very smart GUI toolkit with minimal dependencies and good backwards compat would ensure GTK’s long life and wide adoption. Perhaps they just try to keep it for Gnome-only’s use, viewing it in a competitive way, or they’re deliberately running it into the ground to destroy it and ensure Qt’s adoption (which they’re certainly achieving).

But you have to remember many of these decisions come from corporate and beyond, and are not based on helping Linux. Most of the employees involved just do what they’re told and don’t have a clue.

About

This blog is where you can [check on the latest updates](#) to my Linux software, news and tips, and leave your comments and questions.

[TERMS OF USE](#)

Site info

IgnorantGuru's Blog
[The Andreas04 Theme](#). [Blog at WordPress.com](#).

[Follow](#)

Follow “IgnorantGuru's Blog”

Build a website with [WordPress.com](#)