

Misconduct of GTK+/glib Bugtracker Admins

- From: IgnorantGuru <ignorantguru@openmailbox.org>
- To: gtk-app-devel-list@gnome.org
- Cc: gtk-list@gnome.org
- Subject: Misconduct of GTK+/glib Bugtracker Admins
- Date: Thu, 4 Jun 2015 13:11:55 -0000

Greetings, Not sure where is the best place to bring this - input welcome - but this email is an official complaint against the conduct of https://bugzilla.gnome.org/show_bug.cgi?id=750182 is unacceptable. They are closing bugs preemptively, and are using the threat of censorship to threaten people who discuss the history and background of a bug. They threaten a user with the code of conduct for using the word "overengineered" to justify their choice not to use personally, and they themselves state that people are "not really codeless (sic)". They make false technical statements then delete any corrections offered as a come-as-you-see-it.

You test bugs every day, being run and responded to in an unprofessional and highly biased manner by the Red Hat employees and associated gnome-centric developers, using petty offense at historical projects (allegedly open and free projects, but in which Red Hat employees want portrayed to users and developers? This is not quality maintenance. It is behavior that drives developers away from GTK+ systems. It is behavior that is hardly uncommon - there are many repeats and general complaints about them all over the web.

I have published the following article in my blog which reviews this bug and how it was handled, and participants in the aforementioned thread were not a responding party here by anyone, really no one (merely respond). You can also read all comments there, including those which were deleted.

Please review the conduct of these administrators and participants. It is my understanding that GTK+ and glib are still conducted as open systems, with the understanding that they are used on multiple platforms and systems, and that each have different concerns to be fully heard.

If there is another public forum for complaints against the conduct of these administrators, please direct me there or bring this matter to the attention. Thank you for reviewing this matter, and I look forward to your timely response. My blog report on this incident, in which you can read glib deleted comments, is provided below.

IgnorantGuru: <https://gurblog.wordpress.com/2015/06/04/red-hat-gnome-developers-censoring-gtk-bugtracker/>
URL: https://bugzilla.gnome.org/show_bug.cgi?id=750182
Posted: 4 June 4, 2015

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.

As I have said before, there are many reasons why I do not use this project merely for using the word "overengineered" (lol - if the participants... I thought it might be valuable to bring what they thought to target attention to.

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 is not the only sphere where there can be no open discussion of larger issues facing GTK+. But I don't have to kiss ass, and they have never come doing anything useful in response to bugs I have tried to get them to ignore (e.g. GVFS will fall back to XDG_CACHE_HOME, since it's where GVFS mounted lib). 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 GUT too kit.

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 in the project, more like (RED) 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 I have seen directly 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 "Enlight" a huge debate about gvfs dependency and usdiks, which has nothing to do with this simple bug. As such, they are obstructing, not resolving anything. When I have pointed out gvfs dependency information is pointed out, they delete the message. Further, it's revealed that it's broken in the first place because someone inserted a hack for gvfs into glib, breaking the support 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. Emanuele 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.

GNOME Bugzilla - Bug 750182 - Bug 750182 - 2015-06-02 16:18:25 UTC
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

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

Status: REOPENED

Product: glib

Component: gio

Assigned to: gtkdev

Description Psy[H] 2015-06-31 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.

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 network filesystems like 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 aren't properly 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 users. 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 I have installed gvfs successfully finds mountpoints in /media, /run/media/SUSER, \$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 I have installed gvfs successfully finds mountpoints in /media, /run/media/SUSER, \$HOME/ (excluding hidden dirs).

GIO has code to find mounts. I don't know if it avoids hidden directories.

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 can't move this bug to glib - I'm not 100% sure what heuristics GIO mounts.c applies when looking for mounts.

Comment 7 Emanuele Bassi 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 I have installed gvfs successfully finds mountpoints in /media, /run/media/SUSER, \$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 \$HOME/.cache. 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 they're not, they're not really interop, and they're instead spreading heavy-handed censorship and threats to yield timely replies.

(In reply to Matthias Clasen from comment #6)

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

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/SUSER. 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 Emanuele Bassi 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/SUSER, 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/SUSER.

The bug in GIO is that the code doing the check hardcodes /run instead of /run/media/SUSER, which is not what XDG_RUNTIME is. That's a fallback to XDG_CACHE_DIR (which is what g_get_user_runtime_dir() returns), but that would mean that the user mount points would go under /run/media/SUSER, which is a bit ridiculous.

One of my suggestions 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/FAIK is usually created by udisks, and here are two difficulties:

- udisks moved mounts to /media/SUSER/

- udisks is not installed on every system.

Using /run is not user-writable, so there could be no tool to create /run/media/SUSER 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 /run.

Comment 12 Psy[H] 2015-06-02 17:38:57 UTC

As far as I can tell, XDG_RUNTIME_DIR is typically resolved into /run/user/SUID 700 permissions are mandatory by the spec.

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

/run/user/SUID is a tmpfs mount by itself. Maybe that is why GTKFileChooser does not see those mounts inside, despite it looks like it's installed.

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 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 the traditional use. If you merely support the file manager, 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 much you insist.

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 is 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, it's not helping referring to non-existent variables. eg. <https://bugzilla.gnome.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 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)."?

That gio included a gvfs-specific hack, breaking freedesktop. If g_get_user_runtime_dir() falls back to XDG_CACHE_HOME, that should be searched nowhere in freedesktop 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 fall back to ~/.cache on Debian at least. In fact this is probably why mounts have been placed there for years by many file managers, and it's 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/SUSER, 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 include gvfs 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 searched 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 this location is commonly used. The whole point of mount-point discovery is convenience, not contrived spec compliance (and hacks) to interoperate with it only works with a small subset of systems.

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/SUSER on some systems, it's not 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 noexec restrictions. There are reasons why file managers use XDG_RUNTIME_DIR as a more reasonable default, and why the chooser should eventually be deleted. Personally, I've never seen such mounts in /run, except for root-mounted things like the old udisks behavior (which was eventually all 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 Emanuele Bassi 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 mount_guess should display(). I'm not a GID maintainer, though, so you will need somebody else's ACK for it.

(In reply to Psy[H] from comment #10)

- udisks moved mounts to /media/SUSER/

- 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 you're creating the /run/media/SUSER directory.

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 you're not to have". The reason the dependency is "soft" (i.e. we depend some libraries) is mostly a case of 1. historical and 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/SUSER 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/SUSER in the system.

Anything that creates/run can also create /run/media/SUSER 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 2015-06-02 21:58:19 UTC

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 you're creating the /run/media/SUSER directory.

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 you're not to have". The reason the dependency is "soft" (i.e. we depend some libraries) is mostly a case of 1. historical and 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/SUSER 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/SUSER in the system.

Anything that creates/run can also create /run/media/SUSER 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 21 IgnorantGuru 2015-06-03 08:25:21 UTC

COMMENT DELETED BY André Klapper

For people who don't know the history on this, modern GTK devs (low Red Hat - some of the same names we see here) tried desperately to make GTK work everywhere, so they were forced to back track. To no one's surprise but theirs, GTK still runs just fine without gvfs (except for the few things that inevitably break it, like this gvfs hack interfering with other software), but I'm sure it's still their agenda to make it a gvfs-only system, 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 don'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 your code, and we'll accept it" etc. before from these same people. That's the only way you're just ignoring you - they'd rather waste your time than tell you that they refuse to support you (or really GTK), and won't accept any changes that do so. At least they seem to have 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/SUSER anymore anyway.

XDG_RUNTIME_DIR is not user-owned, so it's no good for fuse.

XDG_RUNTIME_DIR also seem to be only maintained by system-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 system-related stuff like lib like udisks, gvfs, system, etc. XDG_CACHE_HOME or \$HOME/.cache is the only 100% backwards compatible fallback.

Comment 23 Emanuele Bassi 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 (low 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 Emanuele Bassi 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 completely lose functionality of gvfs directly inside its code base is of experience 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/SUSER anymore anyway.

Nevertheless, /run/media/SUSER 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 system-related components which is not universal.

It should be universal, and it's not system-related: that's why it's in the basedir specification on f.d.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>

So, I'll 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 glibby 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 lib like udisks, gvfs, system, etc.

Please, refrain from making comments like this in the future. The things like gvfs or udisks are complex systems it's 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>.

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