

Q Search

GNOME Home Mailing Lists List Archives MAIL SERVICES

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 -0600

Greetings. Not sure where is the best place to bring this - input welcome - but this email is an official complaint against the conduct of:

Matthias Clasen Emmanuele Bassi André Klapper

Specially, their conduct administering the GTK+ bugtracker in the case of bug https://bugzilla.gnome.org/show_bug.cgi?id=750182 is unacceptable. They are closing bugs preemptively, and are using the code of conduct to threaten people who discuss the history and the 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 systemd. They claim personal attacks when no one was even referred to personally, and they themselves state that people are "not really cool" (Emmanuele Bassi). They make false technical statements then delete any corrections offered in comments.

Your GTK+ bug tracker is being run and responded to in an unprofessional and highly biased manner by these Red Hat employees and associated Gnome-centric developers, using petty offense at historical and technical discussions as grounds for threats. Is this really the image you want portrayed to users and developers? This is not quality maintenance. It is behavior that drives developers away from GTK+. And this is hardly uncommon - there are many repeats and general complaints about them all over the web.

I have published the following article on my blog which reviews this bug and how it was handled, and participants in the aforementioned report are welcome to respond there as well (I for one do not censor, I 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 projects, 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 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 all deleted comments, is provided below.

IgnorantGuru

URL

h<u>ttp://igurublog.wordpress.com</u> (Note: I am only subscribed to gtk-app-devel-list gnome org so please CC other replies to me - thanks.) : Red Hat / Gnome Developers Censoring GTK+ Bugtracker Post

https://igurublog.wordpress.com/2015/06/04/red-hat-gnome-developers-censoring-gtk-bugtracker/
Posted : 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. Since they deleted several of my comments, and threatened another participant merely for using the word "overengineered" (lol - if th shoe fits...), I thought it might be valuable to bring what they deleted to larger attention. if the

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 vehatover spaces administratively, for whatever reason.

Here is the full thread, with comments deleted by Red Tape restored. GNOME Bugzilla — Bug 750182 (<u>https://bugzilla.gnome.org/show_bug.cgi?id=750182</u>)

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.

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

I realize that GTK development is effectively dead for non-Gnome 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 by Gnome/Freedesktop. 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 gv1 s to

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). 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'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 <u>https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=736448</u> 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

Iow, 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 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 overengieered 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 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 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 overengieered 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 <u>https://wiki.gnome.org/CodeOfConduct</u> .

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 occassionally 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 arudae here)

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.	
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	
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 <u>http://xkcd.com/1357/</u> 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 irrelevent 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	

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

Add a comment to this post: https://igurublog.wordpress.com/2015/06/04/red-hat-gnome-developers-censoring-gtk-bugtracker/#respond

Follow-Ups:

Re: Misconduct of GTK+/glib Bugtracker Admins • From: John Tall

[Date Prev][Date Next] [Thread Prev][Thread Next] [Thread Index] [Date Index] [Author Index]

The GNOME Project About Us Get Involved Teams Support GNOME Contact Us The GNOME Foundation Resources

Developer Center Documentation Wiki Mailing Lists IRC Channels Bug Tracker Development Code Build Tool News

Press Releases Latest Release Planet GNOME Development News Identi.ca Twitter Copyright © 2005 - 2013 The GNOME Project.

Optimised for standards. Hosted by Red Hat. Powered by MailMan, Python and GNU.