

Apport

- [1. Bugs in apport](#)
- [2. Improvements in Cache and GUI](#)
 - [2.1. Implementations](#)
 - [2.2. Result](#)
- [3. Debian BTS Integration](#)
- [4. Reportbug Integration](#)

1. Bugs in apport

<https://bugs.debian.org/cgi-bin/pkgreport.cgi?ordering=normal;archive=0;src=apport;dist=unstable;repeatmerged=0>

<http://www.researchhut.com/blog/report-bugs-with-apport#comments>
<http://www.researchhut.com/blog/apport-on-debian>

2. Improvements in Cache and GUI

We would like to make two improvements:

- Add another button in the message dialog that shows up when a crash cannot be reported
- `~/.cache/apport` takes too much disk space

[rrs] I meant that apport should check into cachedir (i.e. system cache `/var/cache/apt/archives/`) before downloading. This way, a lot of network bandwidth can be saved.

Secondly, I don't think the idea that the cache should be preserved per user in `~/.cache/apport/`. So If we implement it to consult the system cache, then we can propose to knockoff the user cache completely.

`/var/cache/apt/archives` and `~/.cache/apport/` have different structures. As we can see from the two figures below, the former only contains deb packages, however the latter has additional files aside from deb packages.

```

yurushao@debian:~/cache/apport/retrace$ tree
.
├── Debian GNU
│   └── Linux 7
│       ├── wheezy-Contents-amd64.gz
│       ├── wheezy-proposed-Contents-amd64.gz
│       ├── wheezy-security-Contents-amd64.gz
│       └── wheezy-updates-Contents-amd64.gz
└── system
    └── apt
        └── etc
            └── apt
                ├── apt.conf.d
                ├── preferences.d
                ├── sources.list
                ├── sources.list.d
                ├── trusted.gpg
                └── trusted.gpg.d
                    ├── debian-archive-jessie-automatic.gpg
                    ├── debian-archive-jessie-security-automatic.gpg
                    ├── debian-archive-jessie-stable.gpg
                    ├── debian-archive-squeeze-automatic.gpg
                    ├── debian-archive-squeeze-stable.gpg
                    ├── debian-archive-wheezy-automatic.gpg
                    └── debian-archive-wheezy-stable.gpg

yurushao@debian:/var/cache/apt$ tree
.
├── archives
│   ├── accountsservice_0.6.37-3+b1_amd64.deb
│   ├── acl_2.2.52-2_amd64.deb
│   ├── acpi_1.7-1_amd64.deb
│   ├── acpid_1%3a2.0.23-2_amd64.deb
│   ├── acpi-fakekey_0.142-6_amd64.deb
│   ├── acpi-support_0.142-6_all.deb
│   ├── acpi-support-base_0.142-6_all.deb
│   ├── adwaita-icon-theme_3.14.0-2_all.deb
│   ├── adwaita-icon-theme_3.16.2.1-2_all.deb
│   ├── aglfn_1.7-3_all.deb
│   ├── aisleriot_1%3a3.14.1-1_amd64.deb
│   ├── aisleriot_1%3a3.16.2-1_amd64.deb
│   ├── akonadi-backend-mysql_1.13.0-2_all.deb
│   └── akonadi-server_1.13.0-2_amd64.deb
└── apt
    └── etc
        └── apt
            ├── apt.conf.d
            ├── preferences.d
            ├── sources.list
            ├── sources.list.d
            ├── trusted.gpg
            └── trusted.gpg.d
                ├── debian-archive-jessie-automatic.gpg
                ├── debian-archive-jessie-security-automatic.gpg
                ├── debian-archive-jessie-stable.gpg
                ├── debian-archive-squeeze-automatic.gpg
                ├── debian-archive-squeeze-stable.gpg
                ├── debian-archive-wheezy-automatic.gpg
                └── debian-archive-wheezy-stable.gpg

```

So to save disk space and bandwidth, I think it's better to create symbolic links. For a deb package we're gonna download, we check if it exists in system cache, i.e., /var/cache/apt. If so, we don't need to download it again -- just create a soft link, as the figure below showing:

```

var
├── cache
│   └── apt
│       └── archives
│           ├── coreutils_8.13-3.5_amd64.deb
│           ├── dpkg_1.17.25_amd64.deb -> /var/cache/apt/archives/dp
│           ├── gcc-4.9-base_4.9.2-16_amd64.deb -> /var/cache/apt/archi
│           ├── libacl1_2.2.51-8_amd64.deb
│           ├── libattr1_1%3a2.4.46-8_amd64.deb
│           ├── libbz2-1.0_1.0.6-8_amd64.deb -> /var/cache/apt/arch:
│           ├── libc6_2.19-18_amd64.deb -> /var/cache/apt/archives/
│           └── libc6-dbg_2.19-18_amd64.deb -> /var/cache/apt/archi

```

As far as I have tested, this approach works fine.

[rrs] This may not work when those paths are across different partitions and file systems. Symbolic links don't traverse across file systems.

I think you have already cracked it well. All we want is that our .deb packages are reused. So, just add the logic to check in /var/cache/apt/archives/ for the respective deb, before downloading it from the internet. If it is in the cache, then just copy it from there to the user cache dir ~/.cache/apport.

We may need to find a mechanism to keep the user cache clean. But that is something that can be looked later.

2.1. Implementations

When a crash is found unreportable, Apport just shows a dialog with only “Close” button. To help users address the problem more quickly, we could:

1. Add a “Return” button to return to the main dialog
2. Add an “Install” button to install debugging symbols

[rrs] Let’s have both the options available. Some users may want to download the symbols and report the bugs, while others may simply want to ignore it, or have a look at it at some other time.

ui.py provides an abstract class `UserInterface` for different UI implementations (i.e., GTK, KDE, command line). For GTK environment, `GTKUserInterface` in `apport-gtk.py` inherits from `UserInterface`. The function call sequence when we run `$apport-bug /var/crash/xxx.crash`:

```
app = GTKUserInterface()
app.run_argv() [implemented in ui.py]
→ run_crash()
→ apport.fileutils.mark_report_seen(crash_file)
→ load_report(crash_file) [load report and do consistency checks]
→ [check if current computer has enough memory]
→ apport.fileutils.allowed_to_report()
→ ui_present_report_details() [the main UI dialog]
→ apport.fileutils.mark_report_upload()
→ check_unreportable() [show a dialog if unreportable, our target!]
```

ui.py#check_unreportable():

```
def check_unreportable(self):
    '''Check if the current report is unreportable.
    If so, display an info message and return True.
    '''
    if not self.crashdb.accepts(self.report):
        return False
    if 'UnreportableReason' in self.report:
        if type(self.report['UnreportableReason']) == bytes:
            self.report['UnreportableReason'] =
self.report['UnreportableReason'].decode('UTF-8')
        if 'Package' in self.report:
            title = _('Problem in %s') % self.report['Package'].split()[0]
        else:
            title = ''

    # The default message dialog only has one button
    # We need to customize a dialog to fulfill our requirement
```

```

        #self.ui_info_message(title, _('The problem cannot be
reported:\n\n%s') % self.report['UnreportableReason'])

        self.ui_unreportable_message(title, _('The problem cannot be
reported:\n\n%s') % self.report['UnreportableReason'])

        return True
    return False

```

apport-gtk#ui_unreportable_message():

```

def ui_unreportable_message(self, title, text):
    '''
    This message dialog is used to request user actions when a crash
    cannot be reported
    '''
    self.umd = Gtk.MessageDialog(type = Gtk.MessageType.ERROR, \
        buttons = Gtk.ButtonsType.OK_CANCEL)

    # the logic of displaying text is copied from the original impl
    if 'http://' in text or 'https://' in text:
        if type(text) != type(b''):
            text = text.encode('UTF-8')
        text = GLib.markup_escape_text(text)
        text = re.sub(r'(https?://[a-zA-Z0-9._-]+(?:[a-zA-Z0-9_#?%/-])*)',
            r'<a href="\1">\1</a>', text)
        self.umd.set_markup(text)
    else:
        self.umd.set_property('text', text)
    self.umd.set_title(title)

    # retrieve button objects so that we can customize them
    btnbox = self.umd.vbox.get_children()[1]
    btn_cancel = btnbox.get_children()[0].get_children()[0]
    btn_ok = btnbox.get_children()[0].get_children()[1]

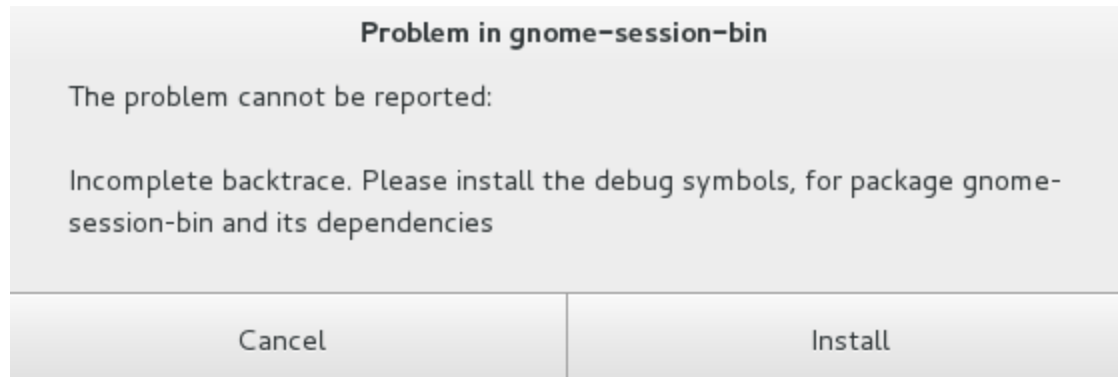
    # set new labels
    btn_cancel.set_label('Cancel')
    btn_ok.set_label('Install')

    # show a tooltip
    btn_ok.set_tooltip_text('Click this to install required debugging
symbols')
    response = self.umd.run()
    if response == Gtk.ResponseType.OK:
        self.umd.destroy()
        self.examine()
    elif response == Gtk.ResponseType.CANCEL:
        pass

```

2.2. Result

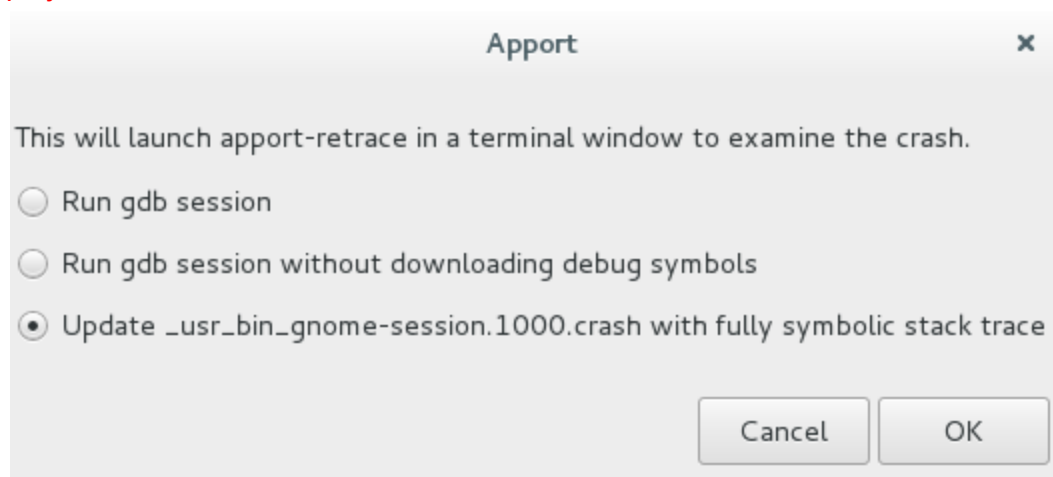
Now there are two buttons in the unreportable message dialog.



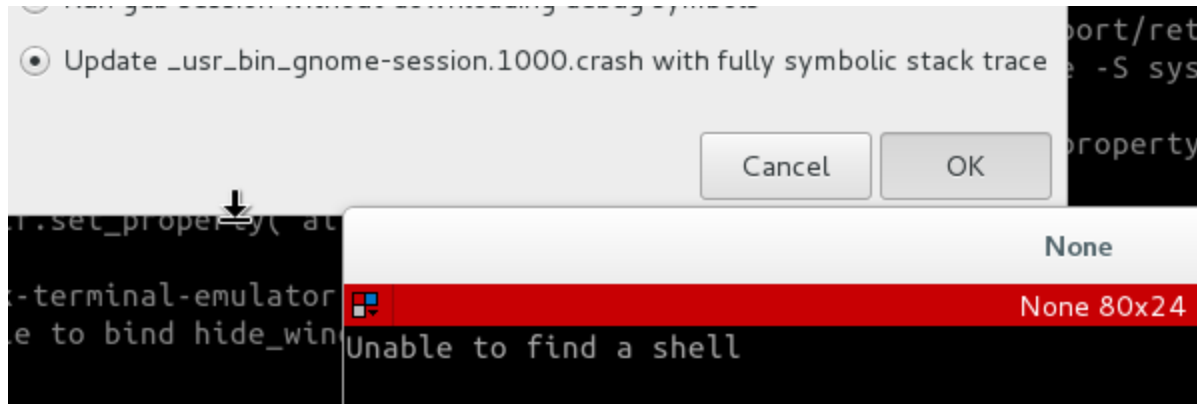
If “Install” is clicked, apport-retrace dialog shows up. In this case, the 3rd radio button is selected by default.

[rrs] This looks good. What will happen next, is something you can display as a tooltip when the user hovers over the “Install” button.

As for consistency, you may be right. But those fine UI perfections can be left for the end of the project.



A new bug (Terminal will be started but reports “unable to find a shell”):



[rrs] Umm.. Have a look at gtk/apport-gtk.

code:

```
terminals = ['x-terminal-emulator', 'gnome-terminal', 'terminator',
```

x-terminal-emulator is a UI independent way to fire the right terminal in Debian, and its derivatives.

In fact, for kde/apport-kde, they weren't doing the same. So I had added a similar patch: [debian/patches/kde-retrace-support.patch](#)

[FIXED]

This bug is similar to the problem reported at:

<https://bugs.launchpad.net/terminator/+bug/1420829>. When we call subprocess.call() function to start a terminal, the code we use looks like:

```
subprocess.call(['terminator', '-e', "apport-retrace -S system -C ..."])
```

Terminator treats all things quoted in the 3rd parameter as a whole. Therefore it cannot find a command called "apport-retrace -S system -C ...".

To solve this we should split the 3rd parameter into single parts, so change to code to:

```
subprocess.call(['terminator', '-e', 'apport-retrace', '-S', 'system', '-C', ...])
```

[rrs] Cool.

3. Debian BTS Integration

<https://github.com/venthur/python-debianbts>

<https://packages.qa.debian.org/p/python-debianbts.html>

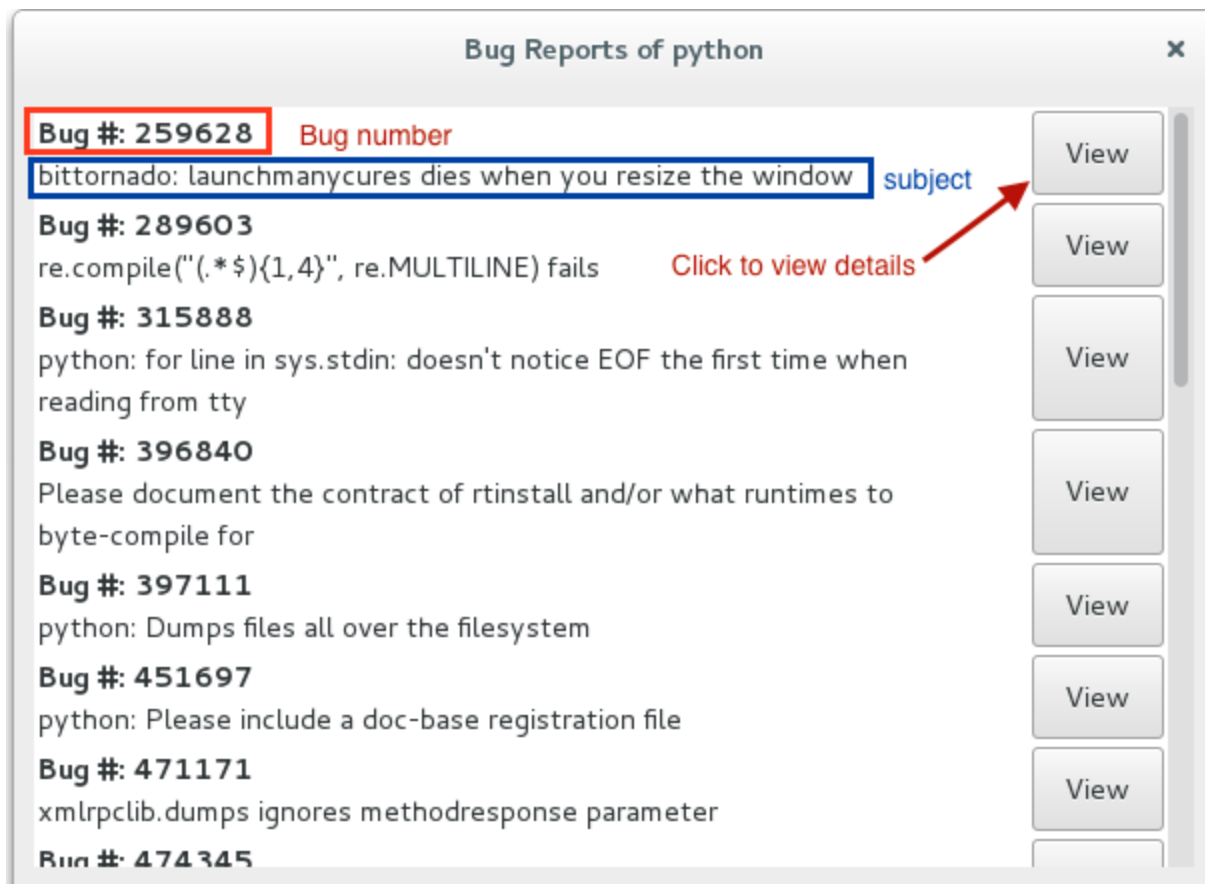
Package name: python-debianbts

Commit ID: 02b7a5239730d25b4ee7a3f51db5ff72cdc71893

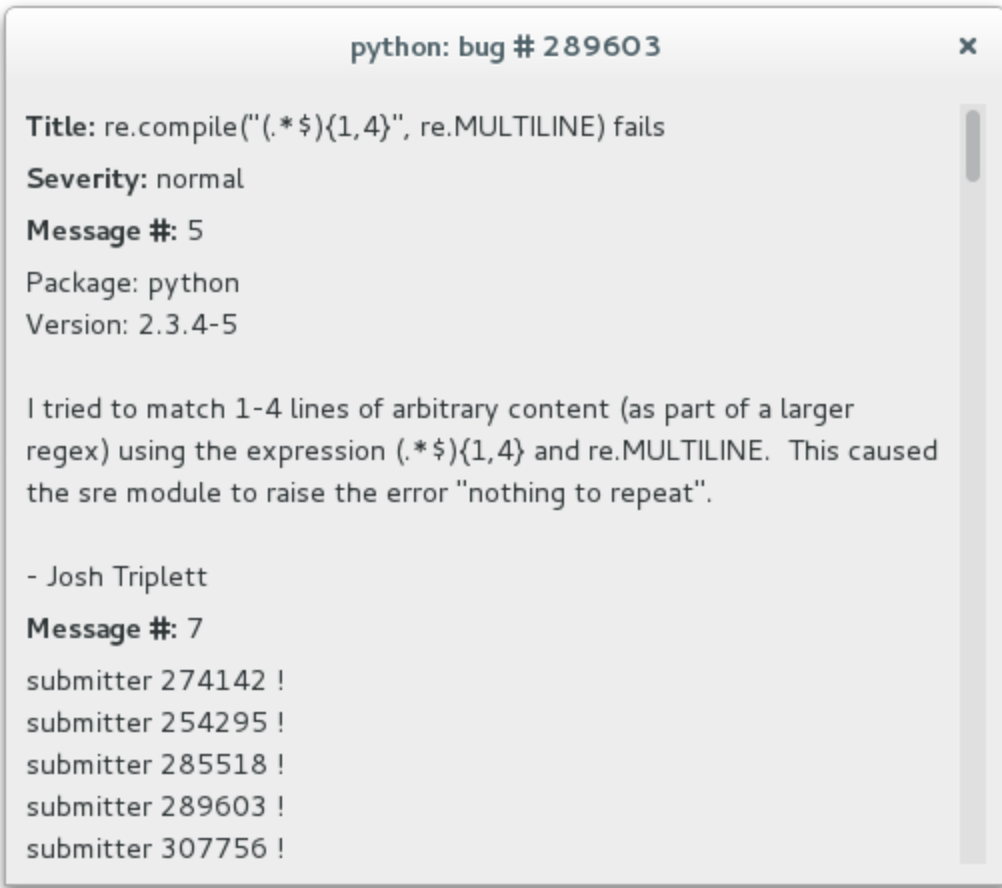
I implemented a window that can show all bug reports in Debian BTS and let users view report details: https://github.com/windflyer/apport/blob/gsoc/gtk/display_bugreports.py

To test this window, we can run it directly, providing a package name as the only parameter:
\$ python display_bugreports.py python (this will show bug reports of the python package)

Then a window will show up. It displays a list of reported bugs in a listbox widget.



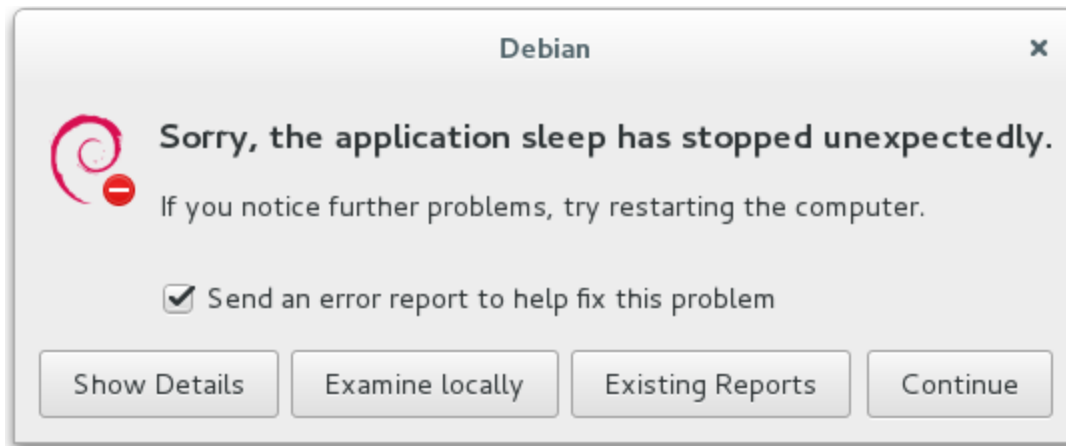
Each row of the listbox contains three components: (1) bug number (2) bug report subject and (3) a "View" button. If the user clicks "View" a new window showing bug report details will be displayed.



Currently I only added (1) title (2) severity (3) a list of messages in this window. But it is easy to add more, for example, originator, reported date, tags, etc.

[rrs] It looks good. Just keep all the data type, that is being displayed, as strings. So that if tomorrow we need to display more data, we don't really have to fiddle with the UI.

Next I would like to add a button in apport-gtk which let the user browse existing bug reports in Debian BTS. I added an "Existing Reports" button in the main dialog of apport-gtk.



To achieve this, I add the following code into `apport-gtk.ui`:

```
<!-- click to show existing reports in Debian BTS -->
<child>
  <object class="GtkButton" id="existing_reports">
    <property name="label" translatable="no">Existing Reports</property>
    <property name="visible">True</property>
    <property name="can_focus">True</property>
    <property name="receives_default">True</property>
    <property name="use_action_appearance">False</property>
    <property name="use_underline">True</property>
    <property name="tooltip_text">Fetch and browse existing bug reports in Debian BTS</property>
    <signal name="clicked" handler="on_existing_reports_clicked" swapped="no"/>
  </object>
  <packing>
    <property name="expand">False</property>
    <property name="fill">True</property>
    <property name="position">2</property>
  </packing>
</child>
```

Finally, I have to set up an event handler for the newly added button “Existing Reports”, and show new windows in `display_bugreports.py`. This step should be fairly easy and straightforward. However, I encountered a problem in python versions.

On my machine, Apport is using python 3.4.2. But it seems that `python-debianbts` doesn't support python 3 at this point. So when I try to import `debianbts`, the `ImportError` happens:

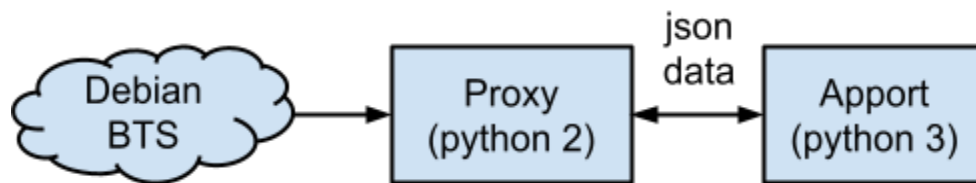
```
Traceback (most recent call last):
  File "/usr/share/apport/apport-gtk", line 629, in on_existing_reports_clicked
    import display_bugreports
  File "/usr/share/apport/display_bugreports.py", line 3, in <module>
    import debianbts
ImportError: No module named 'debianbts'
```

[rrs] Sigh!!!. <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=732644>

We are out of luck here. I do not see any clear plans on how this will be tackled. Your best bet, for now, will be to extract the data using the python2 module, write it to a file, and then use a parser in python3 to interpret it.

If you have any other ideas, I'd like to hear them too ?

I just came up with an idea: we assume both python 2 and python 3 are installed. So we can have a daemon (implemented in python 2) served as a proxy and let Apport (python 3) request bug report data from that daemon. For IPC (the daemon and Apport) we can use socket, e.g., Unix domain socket.



4. Reportbug Integration

reportbug
python-reportbug

`/usr/share/bug/`: where applications put their scripts for retrieving crash information.

crash -> dialog -> report bug -> existing bugs -> hasn't been reported -> submit new -> call reportbug module and report the bug

the final file should be like an email

crashdb_impl and reportbug provide similar functionality, let the user choose which one to use

How to prioritize bug reports that are shown to users?
we only care about bug reports filed through apport?
give me all bug reports which have the User tag ``apport''.

Why don't we let apport call reportbug directly, i.e.,
`subprocess.call(['reportbug', 'pkgname'])`

Problem when import reportbug modules:

```

yurushao@debianvm:~/TMP$ ./reportbug
Gtk-Message: Failed to load module "canberra-gtk-module"
Traceback (most recent call last):
  File "./reportbug", line 35, in <module>
    import gtk2_ui
  File "/home/yurushao/TMP/gtk2_ui.py", line 50, in <module>
    from reportbug import debbugs
  File "/usr/lib/python2.7/dist-packages/reportbug/debbugs.py", line 24, in <module>
    import utils
  File "/usr/lib/python2.7/dist-packages/reportbug/utils.py", line 72, in <module>
    import ui.text_ui as ui
  File "/usr/lib/python2.7/dist-packages/reportbug/ui/text_ui.py", line 38, in <module>
    from reportbug import debbugs, hiermatch
ImportError: cannot import name debbugs

```

Full UI (customized reportbug):

utils.py: exec_and_parse

crashdb_impl.py: show both apport and reportbug collected info

/usr/share/bug/ -- hooks

```

('body:', u"Package: gedit\nVersion: 3.14.0-3\nSeverity: minor\nTags: newcomer\n\nDear
Maintainer,\n\n*** Reporter, please consider answering these questions, where appropriate
***\n\n * What led up to the situation?\n * What exactly did you do (or not do) that was
effective (or\n ineffective)?\n * What was the outcome of this action?\n * What outcome did
you expect instead?\n\n*** End of the template - remove these template lines ***\n\nhello\n\n--
Package-specific info:\nActive plugins:\n - 'time'\n - 'modelines'\n - 'spell'\n - 'docinfo'\n -
'filebrowser'\n\nNo plugin installed in $HOME.\n\nModule versions:\n - glib 2.42.1\n -
gtk+ \n - gtksourceview \n - pygobject \n - enchant \n -
iso-codes 3.57\n\n-- System Information:\nDebian Release: 8.1\n APT prefers
stable-updates\n APT policy: (500, 'stable-updates'), (500, 'stable'), (1,
'experimental')\nArchitecture: amd64 (x86_64)\nKernel: Linux 3.16.0-4-amd64 (SMP w/1 CPU
core)\nLocale: LANG=en_US.UTF-8, LC_CTYPE=en_US.UTF-8 (charmap=UTF-8)\nShell:
/bin/sh linked to /bin/dash\nInit: systemd (via /run/systemd/system)\n\nVersions of packages
gedit depends on:\nnii gedit-common 3.14.0-3\nii gir1.2-peas-1.0 1.12.1-2\nii
gnome-icon-theme-symbolic 3.12.0-1\nii gsettings-desktop-schemas 3.14.1-1\nii iso-codes
3.57-1\nii libatk1.0-0 2.14.0-1\nii libc6 2.19-18\nii libcairo-gobject2
1.14.0-2.1\nii libcairo2 1.14.0-2.1\nii libenchant1c2a 1.6.0-10.1\nii
libgdk-pixbuf2.0-0 2.31.1-2+b1\nii libgirepository-1.0-1 1.42.0-2.2\nii libglib2.0-0
2.42.1-1\nii libgtk-3-0 3.14.5-1\nii libgtksourceview-3.0-1 3.14.1-1\nii

```

```

libpango-1.0-0          1.36.8-3\nii libpangocairo-1.0-0      1.36.8-3\nii libpeas-1.0-0
1.12.1-2\nii libx11-6      2:1.6.2-3\nii libxml2          2.9.1+dfsg1-5\nii python3-gi
3.14.0-1\nii python3-gi-cairo  3.14.0-1\nnpn python3:any      <none>\n\nVersions of
packages gedit recommends:\nii yelp  3.14.1-1\nii zenity  3.14.0-1\n\nVersions of packages
gedit suggests:\nii gedit-plugins  3.14.0-2\n\n-- no debconf information\n")
('attachments:', [])
('mua:', None)
('fromaddr:', 'Yuru Shao <shaoyuru@gmail.com>')
('sendto:', '"Debian Bug Tracking System" <submit@bugs.debian.org>')
('ccaddr', None)
('bccaddr', None)
('headers:', [('X-Debbugs-CC', 'shaoyuru@gmail.com'), (u'Subject', u'gedit: test')])
('package:', 'gedit')
('charset:', 'UTF-8')
('mailing:', True)
('sysinfo:', {'specials': {'ftp.debian.org': <function handle_debian_ftp at 0x7fe6c9da81b8>,
'release.debian.org': <function handle_debian_release at 0x7fe6c9da8230>, 'wnpp': <function
handle_wnpp at 0x7fe6c9da82a8>}, 'infofunc': <function debian_infofunc at 0x7fe6c9da8398>,
'name': 'Debian', 'btsroot': 'http://bugs.debian.org/', 'type': 'debbugs', 'nonvirtual': ['linux-image',
'kernel-image'], 'deppkgs': ('gcc', 'g++', 'cpp', 'gcj', 'gpc', 'gobjc', 'chill', 'gij', 'g77', 'python',
'python-base', 'x-window-system-core', 'x-window-system'), 'otherpkgs': {'installation-reports':
'Problems with installing Debian', 'debian-policy': 'Proposed changes in the Debian policy
documentation', 'qa.debian.org': 'Problems related to the quality assurance group', 'general':
'General problems (e.g., that many manpages are mode 755)', 'security.debian.org': 'Problems
with the security updates server', 'cdrom': 'Problems with installation from CD-ROMs',
'buildd.emdebian.org': 'Problems related to building packages for Emdebian',
'release.debian.org': 'Requests regarding Debian releases and release team tools', 'spam':
'Spam (reassign spam to here so we can complain about it)', 'lists.debian.org': 'The mailing lists,
debian-*@lists.debian.org.', 'listarchives': 'Problems with the WWW mailing list archives',
'wiki.debian.org': 'Problems with the Debian wiki', 'mirrors': 'Problems with Debian archive
mirrors.', 'wnpp': 'Work-Needing and Prospective Packages list', 'debian-i18n': 'Requests
regarding Internationalization (i18n) of the distribution', 'ftp.debian.org': 'Problems with the FTP
site and Package removal requests', 'buildd.debian.org': 'Problems and requests related to the
Debian Buildds', 'upgrade-reports': 'Reports of successful and unsuccessful upgrades',
'release-notes': 'Problems with the release notes', 'base': 'General bugs in the base system',
'security-tracker': 'The Debian Security Bug Tracker', 'debian-maintainers': 'Problems and
requests related to Debian Maintainers', 'press': 'Press release issues', 'nm.debian.org': 'New
Maintainer process and nm.debian.org website', 'bugs.debian.org': 'The bug tracking system,
@bugs.debian.org', 'cdimage.debian.org': 'CD Image issues', 'www.debian.org': 'Problems with
the WWW site (including other *.debian.org sites, except alioth)', 'snapshot.debian.org': 'Issues
with the snapshot.debian.org service', 'project': 'Problems related to project administration',
'tech-ctte': 'Issues to be referred to the technical committee'}, 'cgiroot':
'http://bugs.debian.org/cgi-bin/', 'email': '%s@bugs.debian.org'})

```

```
('rtype:', 'debbugs')
('exinfo:', None)
('replyto:', None)
('printonly', None)
('template:', None)
('outfile:', None)
('mta:', '/usr/sbin/sendmail')
('kudos:', False)
('smtp tls:', False)
('smtp host:', 'reportbug.debian.org')
('smtp user:', '')
('smtp passwd:', '')
('paranoid:', False)
('draft path:', None)
('envelope from:', None)
```

```
submit.send_report(
    'Version: 2.17.3-1\nSeverity: minor\nTags: apport\n\ntest reportbug body',
    'Yuru Shao <shaoyuru@gmail.com>',
    '"Debian Bug Tracking System" <submit@bugs.debian.org>',
    None,
    None,
    [('X-Debbugs-CC', 'shaoyuru@gmail.com'), (u'Subject', u'gedit: test')],
    'apport'
    'UTF-8',
    True)
```