Writing FreeBSD Problem Reports

Dag-Erling Smørgrav
Mark Linimon
##: 43184

2013-11-13 # hrs.

##

This article describes how to best formulate and submit a problem report to the FreeBSD Project.

####

1. Introduction

One of the most frustrating experiences one can have as a software user is to submit a problem report only to have it summarily closed with a terse and unhelpful explanation like “not a bug” or “bogus PR”. Similarly, one of the most frustrating experiences as a software developer is to be flooded with problem reports that are not really problem reports but requests for support, or that contain little or no information about what the problem is and how to reproduce it.

This document attempts to describe how to write good problem reports. What, you ask, is a good problem report? Well, to go straight to the bottom line, a good problem report is one that can be analyzed and dealt with swiftly, to the mutual satisfaction of both user and developer.

Although the primary focus of this article is on FreeBSD problem reports, most of it should apply quite well to other software projects.

Note that this article is organized thematically, not chronologically, so you should read through the entire document before submitting a problem report, rather than treat it as a step-by-step tutorial.

2. When to submit a problem report

There are many types of problems, and not all of them should engender a problem report. Of course, nobody is perfect, and there will be times when you are convinced you have found a bug in a program when in fact you have misunderstood the syntax for a command or made a typographical error in a configuration file (though that in itself may sometimes be indicative of poor documentation or poor error handling in the application). There are still many cases where submitting a problem report is clearly *not* the right course of action, and will only serve to frustrate you and the developers. Conversely, there are cases where it might be appropriate to submit a problem report about something else than a bug——an enhancement or a feature request, for instance.

So how do you determine what is a bug and what is not? As a simple rule of thumb your problem is *not* a bug if it can be expressed as a question (usually of the form “How do I do X?” or “Where can I find Y?”). It is not always quite so black and white, but the question rule covers a large majority of cases. If you are looking for an answer, consider posing your question to the FreeBSD general questions ####.

Some cases where it may be appropriate to submit a problem report about something that is not a bug are:

- Requests for feature enhancements. It is generally a good idea to air these on the mailing lists before submitting a problem report.
- Notification of updates to externally maintained software (mainly ports, but also externally maintained base system components such as BIND or various GNU utilities).

  For unmaintained ports (`MAINTAINER` contains `ports@FreeBSD.org`), such update notifications might get picked up by an interested committer, or you might be asked to provide a patch to update the port; providing it upfront will greatly improve your chances that the port will get updated in a timely manner.

  If the port is maintained, PRs announcing new upstream releases are usually not very useful since they generate supplementary work for the committers, and the maintainer likely knows already there is a new version, they have probably worked with the developers on it, they are probably testing to see there is no regression, etc.

  In either case, following the process described in Porter's Handbook will yield the best results.

A bug that can not be reproduced can rarely be fixed. If the bug only occurred once and you can not reproduce it, and it does not seem to happen to anybody else, chances are none of the developers will be able to reproduce it or figure out what is wrong. That does not mean it did not happen, but it does mean that the chances of your problem report ever leading to a bug fix are very slim. To make matters worse, often these kinds of bugs are actually caused by failing hard drives or overheating processors —— you should always try to rule out these causes, whenever possible, before submitting a PR.

Next, to decide to whom you should file your problem report, you need to understand that the software that makes up FreeBSD is composed of several different elements:

- Code in the base system that is written and maintained by FreeBSD contributors, such as the kernel, the C library, and the device drivers (categorized as `kern`); the binary utilities (`bin`); the manual pages and documentation (`docs`); and the web pages (`www`). All bugs in these areas should be reported to the FreeBSD developers.
- Code in the base system that is written and maintained by others, and imported into FreeBSD and adapted. Examples include bind, gcc(1), and sendmail(8). Most bugs in these areas should be reported to the FreeBSD developers; but in some cases they may need to be reported to the original authors instead if the problems are not FreeBSD-specific. Usually these bugs will fall under either the `bin` or `gnu` categories.
- Individual applications that are not in the base system but are instead part of the FreeBSD Ports Collection (category `ports`). Most of these applications are not written by FreeBSD developers; what FreeBSD provides is merely a framework for installing the

application. Therefore, you should only report a problem to the FreeBSD developers when you believe the problem is FreeBSD-specific; otherwise, you should report it to the authors of the software.

Then you should ascertain whether or not the problem is timely. There are few things that will annoy a developer more than receiving a problem report about a bug she has already fixed.

If the problem is in the base system, you should first read the FAQ section on FreeBSD versions, if you are not already familiar with the topic. It is not possible for FreeBSD to fix problems in anything other than certain recent branches of the base system, so filing a bug report about an older version will probably only result in a developer advising you to upgrade to a supported version to see if the problem still recurs. The Security Officer team maintains the list of supported versions.

If the problem is in a port, note that you must first upgrade to the latest version of the Ports Collection and see if the problem still applies. Due to the rapid pace of changes in these applications, it is infeasible for FreeBSD to support anything other than the absolute latest versions, and problems with older version of applications simply cannot be fixed.

3. Preparations

A good rule to follow is to always do a background search before submitting a problem report. Maybe your problem has already been reported; maybe it is being discussed on the mailing lists, or recently was; it may even already be fixed in a newer version than what you are running. You should therefore check all the obvious places before submitting your problem report. For FreeBSD, this means:

- The FreeBSD Frequently Asked Questions (FAQ) list. The FAQ attempts to provide answers for a wide range of questions, such as those concerning hardware compatibility, user applications, and kernel configuration.
- The mailing lists——if you are not subscribed, use the searchable archives on the FreeBSD web site. If your problem has not been discussed on the lists, you might try posting a message about it and waiting a few days to see if someone can spot something you have overlooked.
- Optionally, the entire web——use your favorite search engine to locate any references to your problem. You may even get hits from archived mailing lists or newsgroups you did not know of or had not thought to search through.
- Next, the searchable FreeBSD PR database (GNATS). Unless your problem is recent or obscure, there is a fair chance it has already been reported.
- Most importantly, you should attempt to see if existing documentation in the source base addresses your problem.

For the base FreeBSD code, you should carefully study the contents of the /usr/src/UPDATING file on your system or its latest version at http://www.FreeBSD.org/cgi/cvsweb.cgi/src/UPDATING. (This is vital information if you are upgrading from one version to another——especially if you are upgrading to the FreeBSD-CURRENT branch).

However, if the problem is in something that was installed as a part of the FreeBSD Ports Collection, you should refer to /usr/ports/UPDATING (for individual ports) or /usr/ports/CHANGES (for changes that affect the entire Ports Collection). http://www.FreeBSD.org/cgi/cvsweb.cgi/ports/UPDATING and http://www.FreeBSD.org/cgi/cvsweb.cgi/ports/CHANGES are also available via CVSweb.

4. Writing the problem report

Now that you have decided that your issue merits a problem report, and that it is a FreeBSD problem, it is time to write the actual problem report. Before we get into the mechanics of the program used to generate and submit PRs, here are some tips and tricks to help make sure that your PR will be most effective.

4.1. Tips and tricks for writing a good problem report

- *Do not leave the “Synopsis” line empty.* The PRs go both onto a mailing list that goes all over the world (where the “Synopsis” is used for the `Subject:` line), but also into a database. Anyone who comes along later and browses the database by synopsis, and finds a PR with a blank subject line, tends just to skip over it. Remember that PRs stay in this database until they are closed by someone; an anonymous one will usually just disappear in the noise.

- *Avoid using a weak “Synopsis” line.* You should not assume that anyone reading your PR has any context for your submission, so the more you provide, the better. For instance, what part of the system does the problem apply to? Do you only see the problem while installing, or while running? To illustrate, instead of `Synopsis: portupgrade is broken`, see how much more informative this seems: `Synopsis: port sysutils/portupgrade coredumps on -current`. (In the case of ports, it is especially helpful to have both the category and portname in the “Synopsis” line.)

- *If you have a patch, say so.* A PR with a patch included is much more likely to be looked at than one without. If you are including one, put the string `[patch]` at the beginning of the “Synopsis”. (Although it is not mandatory to use that exact string, by convention, that is the one that is used.)

- *If you are a maintainer, say so.* If you are maintaining a part of the source code (for instance, a port), you might consider adding the string `[maintainer update]` at the beginning of your synopsis line, and you definitely should set the “Class” of your PR to `maintainer-update` . This way any committer that handles your PR will not have to check.

- *Be specific.* The more information you supply about what problem you are having, the better your chance of getting a response.

  - Include the version of FreeBSD you are running (there is a place to put that, see below) and on which architecture. You should include whether you are running from a release (e.g. from a CDROM or download), or from a system maintained by cvsup(1) (and, if so, how recently you updated). If you are tracking the FreeBSD-CURRENT branch, that is the very first thing someone will ask, because fixes (especially for high-profile problems) tend to get committed very quickly, and FreeBSD-CURRENT users are expected to keep up.

  - Include which global options you have specified in your `make.conf` . Note: specifying `-O2` and above to gcc(1) is known to be buggy in many situations. While the FreeBSD developers will accept patches, they are generally unwilling to investigate such issues due to simple lack of time and volunteers, and may instead respond that this just is not supported.

  - If this is a kernel problem, then be prepared to supply the following information. (You do not have to include these by default, which only tends to fill up the database, but you should include excerpts that you think might be relevant):

    - your kernel configuration (including which hardware devices you have installed)

    - whether or not you have debugging options enabled (such as `WITNESS`), and if so, whether the problem persists when you change the sense of that option

    - a backtrace, if one was generated

    - the fact that you have read `src/UPDATING` and that your problem is not listed there (someone is guaranteed to ask)

    - whether or not you can run any other kernel as a fallback (this is to rule out hardware-related issues such as failing disks and overheating CPUs, which can masquerade as kernel problems)

  - If this is a ports problem, then be prepared to supply the following information. (You do not have to include these by default, which only tends to fill up the database, but you should include excerpts that you think might be relevant):

    - which ports you have installed

    - any environment variables that override the defaults in `bsd.port.mk` , such as `PORTSDIR`

  - the fact that you have read `ports/UPDATING` and that your problem is not listed there (someone is guaranteed to ask)

- *Avoid vague requests for features.* PRs of the form "someone should really implement something that does so-and-so" are less likely to get results than very specific requests. Remember, the source is available to everyone, so if you want a feature, the best way to ensure it being included is to get to work! Also consider the fact that many things like this would make a better topic for discussion on `freebsd-questions` than an entry in the PR database, as discussed above.

- *Make sure no one else has already submitted a similar PR.* Although this has already been mentioned above, it bears repeating here. It only take a minute or two to use the web-based search engine at `http://www.FreeBSD.org/cgi/query-pr-summary.cgi?query`. (Of course, everyone is guilty of forgetting to do this now and then.)

- *Avoid controversial requests.* If your PR addresses an area that has been controversial in the past, you should probably be prepared to not only offer patches, but also justification for why the patches are "The Right Thing To Do". As noted above, a careful search of the mailing lists using the archives at `http://www.FreeBSD.org/search/search.html#mailinglists` is always good preparation.

- *Be polite.* Almost anyone who would potentially work on your PR is a volunteer. No one likes to be told that they have to do something when they are already doing it for some motivation other than monetary gain. This is a good thing to keep in mind at all times on Open Source projects.

4.2. Before you begin

If you are using the send-pr(1) program, make sure your `VISUAL` (or `EDITOR` if `VISUAL` is not set) environment variable is set to something sensible.

You should also make sure that mail delivery works fine. send-pr(1) uses mail messages for the submission and tracking of problem reports. If you cannot post mail messages from the machine you are running send-pr(1) on, your problem report will not reach the GNATS database. For details on the setup of mail on FreeBSD, see the "Electronic Mail" chapter of the FreeBSD Handbook at `http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/mail.html`.

Make sure that your mailer will not mangle the message on its way to GNATS. In particular, if your mailer automatically breaks lines, changes tabs to spaces, or escapes newline characters, any patch that you submit will be rendered unusable. For the text sections, however, we request that you insert manual linebreaks somewhere around 70 characters, so that the web display of the PR will be readable.

Similar considerations apply if you are using the web-based PR submittal form instead of send-pr(1). Note that cut-and-paste operations can have their own side-effects on text formatting. In certain cases it may be necessary to use uuencode(1) to ensure that patches arrive unmodified.

Finally, if your submission will be lengthy, you should to prepare your work offline so that nothing will be lost in case there is a problem submitting it. This can be an especial problem with the web form.

4.3. Attaching patches or files

The following applies to submitting PRs via email:

The send-pr(1) program has provisions for attaching files to a problem report. You can attach as many files as you want provided that each has a unique base name (i.e. the name of the file proper, without the path). Just use the `-a` command-line option to specify the names of the files you wish to attach:

```
% send-pr -a /var/run/dmesg -a /tmp/errors
```

Do not worry about binary files, they will be automatically encoded so as not to upset your mail agent.

If you attach a patch, make sure you use the `-c` or `-u` option to diff(1) to create a context or unified diff (unified is preferred), and make sure to specify the exact CVS revision numbers of the files you modified so the developers who read your report will be able to apply them easily. For problems with the kernel or the base utilities, a patch against FreeBSD-CURRENT (the HEAD CVS branch) is preferred since all new code should be applied and tested there first. After appropriate or substantial testing has been done, the code will be merged/migrated to the FreeBSD-STABLE branch.

If you attach a patch inline, instead of as an attachment, note that the most common problem by far is the tendency of some email programs to render tabs as spaces, which will completely ruin anything intended to be part of a Makefile.

Do not send patches as attachments using `Content-Transfer-Encoding: quoted-printable`. These will perform character escaping and the entire patch will be useless.

Also note that while including small patches in a PR is generally all right——particularly when they fix the problem described in the PR——large patches and especially new code which may require substantial review before committing should be placed on a web or ftp server, and the URL should be included in the PR instead of the patch. Patches in email tend to get mangled, especially when GNATS is involved, and the larger the patch, the harder it will be for interested parties to unmangle it. Also, posting a patch on the web allows you to modify it without having to resubmit the entire patch in a followup to the original PR. Finally, large patches simply increase the size of the database, since closed PRs are not actually deleted but instead kept and simply marked as `closed`.

You should also take note that unless you explicitly specify otherwise in your PR or in the patch itself, any patches you submit will be assumed to be licensed under the same terms as the original file you modified.

4.4. Filling out the template

The next section applies to the email method only:

When you run send-pr(1), you are presented with a template. The template consists of a list of fields, some of which are pre-filled, and some of which have comments explaining their purpose or listing acceptable values. Do not worry about the comments; they will be removed automatically if you do not modify them or remove them yourself.

At the top of the template, below the `SEND-PR:` lines, are the email headers. You do not normally need to modify these, unless you are sending the problem report from a machine or account that can send but not receive mail, in which case you will want to set the `From:` and `Reply-To:` to your real email address. You may also want to send yourself (or someone else) a carbon copy of the problem report by adding one or more email addresses to the `Cc:` header.

In the email template you will find the following two single-line fields:

- *Submitter-Id:* Do not change this. The default value of `current-users` is correct, even if you run FreeBSD-STABLE.

- *Confidential:* This is prefilled to `no`. Changing it makes no sense as there is no such thing as a confidential FreeBSD problem report——the PR database is distributed worldwide by CVSup.

The next section describes fields that are common to both the email interface and the web interface:

- *Originator:* Please specify your real name, optionally followed by your email address in angle brackets. In the email interface, this is normally prefilled with the `gecos` field of the currently logged-in user.

> *##*
>
> The email address you use will become public information and may become available to spammers. You should either have spam handling procedures in place, or use a temporary email account. However, please note that if you do not use a valid email account at all, we will not be able to ask you questions about your PR.

- *Organization:* Whatever you feel like. This field is not used for anything significant.
- *Synopsis:* Fill this out with a short and accurate description of the problem. The synopsis is used as the subject of the problem report email, and is used in problem report listings and summaries; problem reports with obscure synopses tend to get ignored.

  As noted above, if your problem report includes a patch, please have the synopsis start with `[patch]`; if this is a ports PR and you are the maintainer, you may consider adding `[maintainer update]` and set the "Class" of your PR to `maintainer-update`.
- *Severity:* One of `non-critical`, `serious` or `critical`. Do not overreact; refrain from labeling your problem `critical` unless it really is (e.g. data corruption issues, serious regression from previous functionality in -CURRENT) or `serious` unless it is something that will affect many users (kernel panics or freezes; problems with particular device drivers or system utilities). FreeBSD developers will not necessarily work on your problem faster if you inflate its importance since there are so many other people who have done exactly that —— in fact, some developers pay little attention to this field because of this.

  ##

  Major security problems should *not* be filed in GNATS, because all GNATS information is public knowledge. Please send such problems in private email to Security Officer Team <security-officer@FreeBSD.org>.
- *Priority:* One of `low`, `medium` or `high`. `high` should be reserved for problems that will affect practically every user of FreeBSD and `medium` for something that will affect many users.

  ##

  This field has become so widely abused that it is almost completely meaningless.
- *Category:* Choose an appropriate category.

##

There are a number of "platform" categories into which bugs in the base system that are specific to one particular hardware architecture should be filed. Problems that are generic all across versions of FreeBSD should probably be filed as `kern` or `bin`; see discussion of those categories below.

Example: you have a common PC-based machine, and think you have encountered a problem specific to a particular chipset or a particular motherboard: `i386` is the right category.

Example: You are having a problem with an add-in peripheral card on a commonly seen bus, or a problem with a particular type of hard disk drive: in this case, it probably applies to more than one architecture, and `kern` is the right category.

Here is the current list of categories (taken from `http://www.FreeBSD.org/cgi/cvsweb.cgi/src/gnu/usr.bin/send-pr/categories`):

- `advocacy:` problems relating to FreeBSD's public image. Rarely used.
- `alpha:` problems specific to the Alpha platform.
- `amd64:` problems specific to the AMD64 platform.
- `bin:` problems with userland programs in the base system. If running whereis(1) shows `/bin`, `/usr/sbin`, or something similar, then this is probably the right category. (A few contributed programs might instead need to be in `gnu`; see below.)
- `conf:` problems with configuration files, default values, and so forth. Things that affect `/usr/share` or `/etc/rc*` belong here.
- `docs:` problems with manual pages or on-line documentation.
- `gnu:` problems with imported GNU software such as gcc(1) or grep(1).
- `i386:` problems specific to the i386™ platform.
- `ia64:` problems specific to the ia64 platform.
- `java:` problems related to the Java™ Virtual Machine. (Ports that merely depend on Java™ to run should be filed under `ports`.)

- `kern`: problems with the kernel, (non-platform-specific) device drivers, or the base libraries.
- `misc`: anything that does not fit in any of the other categories. (Note that there is almost nothing that truly belongs in this category, except for problems with the release and build infrastructure. Temporary build failures on `HEAD` do not belong here. Also note that it is easy for things to get lost in this category).
- `ports`: problems relating to the ports tree.
- `powerpc`: problems specific to the PowerPC® platform.
- `sparc64`: problems specific to the Sparc64® platform.
- `standards`: Standards conformance issues.
- `threads`: problems related to the FreeBSD threads implementation (especially on FreeBSD-CURRENT).
- `usb`: problems related to the FreeBSD USB implementation.
- `www`: Changes or enhancements to the FreeBSD website. Problems with code found in `/usr/ports/www` do *not* belong here, they belong in `ports` instead.

• *Class:* Choose one of the following:

- `sw-bug`: software bugs.
- `doc-bug`: errors in documentation.
- `change-request`: requests for additional features or changes in existing features.
- `update`: updates to ports or other contributed software.
- `maintainer-update`: updates to ports for which you are the maintainer.

• *Release:* The version of FreeBSD that you are running. This is filled out automatically if you are using send-pr(1) and need only be changed if you are sending a problem report from a different system than the one that exhibits the problem.

Finally, there is a series of multi-line fields:

• *Environment:* This should describe, as accurately as possible, the environment in which the problem has been observed. This includes the operating system version, the version of the specific program or file that contains the problem, and any other relevant items such as system configuration, other installed software that influences the problem, etc.

——quite simply everything a developer needs to know to reconstruct the environment in which the problem occurs.

- *Description:* A complete and accurate description of the problem you are experiencing. Try to avoid speculating about the causes of the problem unless you are certain that you are on the right track, as it may mislead a developer into making incorrect assumptions about the problem.
- *How-To-Repeat:* A summary of the actions you need to take to reproduce the problem.
- *Fix:* Preferably a patch, or at least a workaround (which not only helps other people with the same problem work around it, but may also help a developer understand the cause for the problem), but if you do not have any firm ideas for either, it is better to leave this field blank than to speculate.

4.5. Sending off the problem report

If you are using send-pr(1):

Once you are done filling out the template, have saved it, and exit your editor, send-pr(1) will prompt you with `s)end, e)dit or a)bort?` . You can then hit **s** to go ahead and submit the problem report, **e** to restart the editor and make further modifications, or **a** to abort. If you choose the latter, your problem report will remain on disk (send-pr(1) will tell you the filename before it terminates), so you can edit it at your leisure, or maybe transfer it to a system with better net connectivity, before sending it with the `-f` to send-pr(1):

```
% send-pr -f ~/my-problem-report
```

This will read the specified file, validate the contents, strip comments and send it off.

If you are using the web form:

Before you hit `submit`, you will need to fill in a field containing text that is represented in image form on the page. This unfortunate measure has had to be adopted due to misuse by automated systems and a few misguided individuals. It is a necessary evil that no one likes; please do not ask us to remove it.

Note that you are `strongly advised` to save your work somewhere before hitting `submit`. A common problem for users is to have their web browser displaying a stale image from its cache. If this happens to you, your submission will be rejected and you may lose your work.

If you are unable to view images for any reason, and are also unable to use send-pr(1), please accept our apologies for the inconvenience and email your problem report to the bugbuster team at <`freebsd-bugbusters@FreeBSD.org`>.

5. Follow-up

Once your problem report has been filed, you will receive a confirmation by email which will include the tracking number that was assigned to your problem report and a URL you can use to check its status. With a little luck, someone will take an interest in your problem and try to address it, or, as the case may be, explain why it is not a problem. You will be automatically notified of any change of status, and you will receive copies of any comments or patches someone may attach to your problem report's audit trail.

If someone requests additional information from you, or you remember or discover something you did not mention in the initial report, please use one of two methods to submit your followup:

- The easiest way is to use the followup link on the individual PR's web page, which you can reach from the PR search page. Clicking on this link will bring up an an email window with the correct To: and Subject: lines filled in (if your browser is configured to do this).
- Alternatively, you can just mail it to <bug-followup@FreeBSD.org >, making sure that the tracking number is included in the subject so the bug tracking system will know what problem report to attach it to.

##

If you do *not* include the tracking number, GNATS will become confused and create an entirely new PR which it then assigns to the GNATS administrator, and then your followup will become lost until someone comes in to clean up the mess, which could be days or weeks afterwards.

Wrong way:

```
Subject: that PR I sent
```

Right way:

```
Subject: Re: ports/12345: compilation problem ↺
with foo/bar
```

If the problem report remains open after the problem has gone away, just send a follow-up (in the manner prescribed above) saying that the problem report can be closed, and, if possible, explaining how or when the problem was fixed.

6. If you are having problems

Most PRs go through the system and are accepted quickly; however, at times GNATS runs behind and you may not get your email confirmation for 10 minutes or even longer. Please try to be patient.

In addition, because GNATS receives all its input via email, it is absolutely vital that FreeBSD runs all its submissions through spam filters. If you do not get a response within an hour or two, you may have fallen afoul of them; if so, please contact the GNATS administrators at <bugmeister@FreeBSD.org > and ask for help.

> ##
>
> Among the anti-spam measures is one that weighs against many common abuses seen HTML-based email (although not necessarily the mere inclusion of HTML in a PR). We strongly recommend against the use of HTML-based email when sending PRs: not only is it more likely to fall afoul of the filters, it also tends to merely clutter up the database. Plain old email is strongly preferred.

On rare occasions you will encounter a GNATS bug where a PR is accepted and assigned a tracking number but it does not show up on the list of PRs on any of the web query pages. What may have happened is that the database index has gotten out of synchronization with the database itself. The way that you can test whether this has happened is to pull up the view a single PR page and see whether the PR shows up. If it does, please notify the GNATS administrators at <bugmeister@FreeBSD.org >. Note that there is a `cron` job that periodically rebuilds the database, so unless you are in a hurry, no action needs to be taken.

7. Further Reading

This is a list of resources relevant to the proper writing and processing of problem reports. It is by no means complete.

- How to Report Bugs Effectively——an excellent essay by Simon G. Tatham on composing useful (non-FreeBSD-specific) problem reports.
- Problem Report Handling Guidelines——valuable insight into how problem reports are handled by the FreeBSD developers.

##

P