NAME

perlbug - how to submit bug reports on Perl

SYNOPSIS

perlbug

perlbug [**-v**] [**-a** *address*] [**-s** *subject*] [**-b** *body* | **-f** *inputfile*] [**-F** *outputfile*] [**-r** *returnaddress*] [**-e** *editor*] [**-c** *adminaddress* | **-C**] [**-S**] [**-t**] [**-d**] [**-A**] [**-h**] [**-T**]

perlbug [**-v**] [**-r** *returnaddress*] [**-A**] [**-ok** | **-okay** | **-nok** | **-nokay**]

perlthanks

DESCRIPTION

This program is designed to help you generate and send bug reports (and thank-you notes) about perl5 and the modules which ship with it.

In most cases, you can just run it interactively from a command line without any special arguments and follow the prompts.

If you have found a bug with a non-standard port (one that was not part of the *standard distribution*), a binary distribution, or a non-core module (such as Tk, DBI, etc), then please see the documentation that came with that distribution to determine the correct place to report bugs.

If you are unable to send your report using **perlbug** (most likely because your system doesn't have a way to send mail that perlbug recognizes), you may be able to use this tool to compose your report and save it to a file which you can then send to **perlbug@perl.org** using your regular mail client.

In extreme cases, **perlbug** may not work well enough on your system to guide you through composing a bug report. In those cases, you may be able to use **perlbug -d** to get system configuration information to include in a manually composed bug report to **perlbug@perl.org**.

When reporting a bug, please run through this checklist:

What version of Perl you are running?

> Type `perl -v` at the command line to find out.

Are you running the latest released version of perl?

> Look at http://www.perl.org/ to find out. If you are not using the latest released version, please try to replicate your bug on the latest stable release.
>
> Note that reports about bugs in old versions of Perl, especially those which indicate you haven't also tested the current stable release of Perl, are likely to receive less attention from the volunteers who build and maintain Perl than reports about bugs in the current release.
>
> This tool isn't appropriate for reporting bugs in any version prior to Perl 5.0.

Are you sure what you have is a bug?

> A significant number of the bug reports we get turn out to be documented features in Perl. Make sure the issue you've run into isn't intentional by glancing through the documentation that comes with the Perl distribution.
>
> Given the sheer volume of Perl documentation, this isn't a trivial undertaking, but if you can point to documentation that suggests the behaviour you're seeing is *wrong*, your issue is likely to receive more attention. You may want to start with **perldoc** *perltrap* for pointers to common traps that new (and experienced) Perl programmers run into.
>
> If you're unsure of the meaning of an error message you've run across, **perldoc** *perldiag* for an explanation. If the message isn't in perldiag, it probably isn't generated by Perl. You may have luck consulting your operating system documentation instead.
>
> If you are on a non-UNIX platform **perldoc** *perlport*, as some features may be unimplemented

or work differently.

You may be able to figure out what's going wrong using the Perl debugger. For information about how to use the debugger **perldoc** *perldebug*.

Do you have a proper test case?

The easier it is to reproduce your bug, the more likely it will be fixed -- if nobody can duplicate your problem, it probably won't be addressed.

A good test case has most of these attributes: short, simple code; few dependencies on external commands, modules, or libraries; no platform-dependent code (unless it's a platform-specific bug); clear, simple documentation.

A good test case is almost always a good candidate to be included in Perl's test suite. If you have the time, consider writing your test case so that it can be easily included into the standard test suite.

Have you included all relevant information?

Be sure to include the **exact** error messages, if any. "Perl gave an error" is not an exact error message.

If you get a core dump (or equivalent), you may use a debugger (**dbx**, **gdb**, etc) to produce a stack trace to include in the bug report.

NOTE: unless your Perl has been compiled with debug info (often **-g**), the stack trace is likely to be somewhat hard to use because it will most probably contain only the function names and not their arguments. If possible, recompile your Perl with debug info and reproduce the crash and the stack trace.

Can you describe the bug in plain English?

The easier it is to understand a reproducible bug, the more likely it will be fixed. Any insight you can provide into the problem will help a great deal. In other words, try to analyze the problem (to the extent you can) and report your discoveries.

Can you fix the bug yourself?

A bug report which *includes a patch to fix it* will almost definitely be fixed. When sending a patch, please use the `diff` program with the `-u` option to generate "unified" diff files. Bug reports with patches are likely to receive significantly more attention and interest than those without patches.

Your patch may be returned with requests for changes, or requests for more detailed explanations about your fix.

Here are a few hints for creating high-quality patches:

Make sure the patch is not reversed (the first argument to diff is typically the original file, the second argument your changed file). Make sure you test your patch by applying it with the `patch` program before you send it on its way. Try to follow the same style as the code you are trying to patch. Make sure your patch really does work (`make test`, if the thing you're patching is covered by Perl's test suite).

Can you use `perlbug` to submit the report?

perlbug will, amongst other things, ensure your report includes crucial information about your version of perl. If `perlbug` is unable to mail your report after you have typed it in, you may have to compose the message yourself, add the output produced by `perlbug -d` and email it to **perlbug@perl.org**. If, for some reason, you cannot run `perlbug` at all on your system, be sure to include the entire output produced by running `perl -V` (note the uppercase V).

Whether you use `perlbug` or send the email manually, please make your Subject line informative. "a bug" is not informative. Neither is "perl crashes" nor is "HELP!!!". These don't help. A compact description of what's wrong is fine.

Can you use `perlbug` to submit a thank-you note?

> Yes, you can do this by either using the `-T` option, or by invoking the program as `perlthanks`. Thank-you notes are good. It makes people smile.

Having done your bit, please be prepared to wait, to be told the bug is in your code, or possibly to get no reply at all. The volunteers who maintain Perl are busy folks, so if your problem is an obvious bug in your own code, is difficult to understand or is a duplicate of an existing report, you may not receive a personal reply.

If it is important to you that your bug be fixed, do monitor the perl5-porters@perl.org mailing list and the commit logs to development versions of Perl, and encourage the maintainers with kind words or offers of frosty beverages. (Please do be kind to the maintainers. Harassing or flaming them is likely to have the opposite effect of the one you want.)

Feel free to update the ticket about your bug on http://rt.perl.org if a new version of Perl is released and your bug is still present.

OPTIONS

-a

Address to send the report to. Defaults to **perlbug@perl.org**.

-A

Don't send a bug received acknowledgement to the reply address. Generally it is only a sensible to use this option if you are a perl maintainer actively watching perl porters for your message to arrive.

-b

Body of the report. If not included on the command line, or in a file with **-f**, you will get a chance to edit the message.

-C

Don't send copy to administrator.

-c

Address to send copy of report to. Defaults to the address of the local perl administrator (recorded when perl was built).

-d

Data mode (the default if you redirect or pipe output). This prints out your configuration data, without mailing anything. You can use this with **-v** to get more complete data.

-e

Editor to use.

-f

File containing the body of the report. Use this to quickly send a prepared message.

-F

File to output the results to instead of sending as an email. Useful particularly when running perlbug on a machine with no direct internet connection.

-h

Prints a brief summary of the options.

-ok

Report successful build on this system to perl porters. Forces **-S** and **-C**. Forces and

supplies values for **-s** and **-b**. Only prompts for a return address if it cannot guess it (for use with **make**). Honors return address specified with **-r**. You can use this with **-v** to get more complete data. Only makes a report if this system is less than 60 days old.

-okay

As **-ok** except it will report on older systems.

-nok

Report unsuccessful build on this system. Forces **-C**. Forces and supplies a value for **-s**, then requires you to edit the report and say what went wrong. Alternatively, a prepared report may be supplied using **-f**. Only prompts for a return address if it cannot guess it (for use with **make**). Honors return address specified with **-r**. You can use this with **-v** to get more complete data. Only makes a report if this system is less than 60 days old.

-nokay

As **-nok** except it will report on older systems.

-r

Your return address. The program will ask you to confirm its default if you don't use this option.

-S

Send without asking for confirmation.

-s

Subject to include with the message. You will be prompted if you don't supply one on the command line.

-t

Test mode. The target address defaults to **perlbug-test@perl.org**.

-T

Send a thank-you note instead of a bug report.

-v

Include verbose configuration data in the report.

AUTHORS

Kenneth Albanowski (<kjahds@kjahds.com>), subsequently *doc*tored by Gurusamy Sarathy (<gsar@activestate.com>), Tom Christiansen (<tchrist@perl.com>), Nathan Torkington (<gnat@frii.com>), Charles F. Randall (<cfr@pobox.com>), Mike Guy (<mjtg@cam.a.uk>), Dominic Dunlop (<domo@computer.org>), Hugo van der Sanden (<hv@crypt.org<gt>), Jarkko Hietaniemi (<jhi@iki.fi>), Chris Nandor (<pudge@pobox.com>), Jon Orwant (<orwant@media.mit.edu>, Richard Foley (<richard.foley@rfi.net>), and Jesse Vincent (<jesse@bestpractical.com<gt>).

SEE ALSO

perl(1), perldebug(1), perldiag(1), perlport(1), perltrap(1), diff(1), patch(1), dbx(1), gdb(1)

BUGS

None known (guess what must have been used to report them?)