NAME

perlbug - how to submit bug reports on Perl

SYNOPSIS

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

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

DESCRIPTION

A program to help generate bug reports about perl or the modules that come with it, and mail them.

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-standard module (such as Tk, CGI, etc), then please see the documentation that came with that distribution to determine the correct place to report bugs.

`perlbug` is designed to be used interactively. Normally no arguments will be needed. Simply run it, and follow the prompts.

If you are unable to run **perlbug** (most likely because you don't have a working setup to send mail that perlbug recognizes), you may have to compose your own report, and email it to **perlbug@perl.org**. You might find the **-d** option useful to get summary information in that case.

In any case, when reporting a bug, please make sure you have 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.com/ to find out. If it is not the latest released version, get that one and see whether your bug has been fixed. Note that bug reports about old versions of Perl, especially those prior to the 5.0 release, are likely to fall upon deaf ears. You are on your own if you continue to use perl1 .. perl4.

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 behavior you are witnessing doesn't fall under that category, by glancing through the documentation that comes with Perl (we'll admit this is no mean task, given the sheer volume of it all, but at least have a look at the sections that *seem* relevant).

Be aware of the familiar traps that perl programmers of various hues fall into. See *perltrap*.

Check in *perldiag* to see what any Perl error message(s) mean. If message isn't in perldiag, it probably isn't generated by Perl. Consult your operating system documentation instead.

If you are on a non-UNIX platform check also *perlport*, as some features may be unimplemented or work differently.

Try to study the problem under the Perl debugger, if necessary. See *perldebug*.

Do you have a proper test case?

The easier it is to reproduce your bug, the more likely it will be fixed, because if no one can duplicate the problem, no one can fix it. A good test case has most of these attributes: fewest possible number of lines; few dependencies on external commands, modules, or libraries; runs on most platforms unimpeded; and is self-documenting.

A good test case is almost always a good candidate to be on the perl test suite. If you have the time, consider making your test case so that it will readily fit into the standard test suite.

Remember also to include the **exact** error messages, if any. "Perl complained something" 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 dump 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. Anything you can provide by way of insight into the problem helps 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. Use the `diff` program to generate your patches (`diff` is being maintained by the GNU folks as part of the **diffutils** package, so you should be able to get it from any of the GNU software repositories). If you do submit a patch, the cool-dude counter at perlbug@perl.org will register you as a savior of the world. Your patch may be returned with requests for changes, or requests for more detailed explanations about your fix.

Here are some clues for creating quality patches: Use the **-c** or **-u** switches to the diff program (to create a so-called context or unified diff). 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 supports it).

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" not informative. Neither is "perl crashes" nor "HELP!!!". These don't help. A compact description of what's wrong is fine.

Having done your bit, please be prepared to wait, to be told the bug is in your code, or even to get no reply at all. The Perl maintainers are busy folks, so if your problem is a small one or if it is difficult to understand or already known, they may not respond with a personal reply. If it is important to you that your bug be fixed, do monitor the `Changes` file in any development releases since the time you submitted the bug, and encourage the maintainers with kind words (but never any flames!). Feel free to resend your bug report if the next released version of perl comes out 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**.

-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>, and Richard Foley (<richard@rfi.net>).

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?)