

NAME

perl - The Perl 5 language interpreter

SYNOPSIS

```
perl [ -sTtuUWX ] [ -hv ] [ -V[:configvar] ] [ -cw ] [ -d[t][:debugger] ] [ -D[number/list] ] [ -pna ] [ -F
pattern ] [ -I[octal] ] [ -O[octal/hexadecimal] ] [ -ldir ] [ -m[-]module ] [ -M[-]'module...' ] [ -f ] [ -C [
number/list] ] [ -S ] [ -x[dir] ] [ -i[extension] ] [ [-e|-E] 'command' ] [ -- ] [ programfile ] [ argument ]...
```

For more information on these options, you can run `perldoc perlrun`.

GETTING HELP

The *perldoc* program gives you access to all the documentation that comes with Perl. You can get more documentation, tutorials and community support online at <http://www.perl.org/>.

If you're new to Perl, you should start by running `perldoc perlintro`, which is a general intro for beginners and provides some background to help you navigate the rest of Perl's extensive documentation. Run `perldoc perldoc` to learn more things you can do with *perldoc*.

For ease of access, the Perl manual has been split up into several sections.

This section is parsed by `Porting/pod_lib.pl` for use by `pod/buildtoc` etc

flag =g perluniprops perlmodlib perlapi perlintern flag =go perltoc flag =ro perlcn perljp perlko perltw
flag = perlvm

path perlfaq.* cpan/perlfaq/lib/ path perlglossary cpan/perlfaq/lib/ path perlxs(?:tut|typemap)?
dist/ExtUtils-ParseXS/lib/ path perldoc cpan/Pod-Perldoc/lib/

aux a2p c2ph h2ph h2xs perlbug pl2pm pod2html pod2man s2p splain xsubpp

Overview

```
perl    Perl overview (this section)
perlintro Perl introduction for beginners
perlrun Perl execution and options
perltoc Perl documentation table of contents
```

Tutorials

```
perlreftut  Perl references short introduction
perldsc     Perl data structures intro
perllo1     Perl data structures: arrays of arrays

perlrequick Perl regular expressions quick start
perlretut   Perl regular expressions tutorial

perllootut  Perl OO tutorial for beginners

perlperf    Perl Performance and Optimization Techniques

perlstyle   Perl style guide

perlcheat   Perl cheat sheet
perltrap    Perl traps for the unwary
perldebtut  Perl debugging tutorial

perlfaq     Perl frequently asked questions
```

perlfaq1 General Questions About Perl
perlfaq2 Obtaining and Learning about Perl
perlfaq3 Programming Tools
perlfaq4 Data Manipulation
perlfaq5 Files and Formats
perlfaq6 Regexes
perlfaq7 Perl Language Issues
perlfaq8 System Interaction
perlfaq9 Networking

Reference Manual

perlsyn Perl syntax
perldata Perl data structures
perlop Perl operators and precedence
perlsub Perl subroutines
perlfunc Perl built-in functions
 perlopentut Perl open() tutorial
 perlpacktut Perl pack() and unpack() tutorial
perlpod Perl plain old documentation
perlpodspec Perl plain old documentation format specification
perlpodstyle Perl POD style guide
perldiag Perl diagnostic messages
perllexwarn Perl warnings and their control
perldebug Perl debugging
perlvar Perl predefined variables
perlre Perl regular expressions, the rest of the story
perlrebackslash Perl regular expression backslash sequences
perlrecharclass Perl regular expression character classes
perlref Perl regular expressions quick reference
perlref Perl references, the rest of the story
perlform Perl formats
perlobj Perl objects
perltye Perl objects hidden behind simple variables
 perlshmfilter Perl DBM filters

perlipc Perl interprocess communication
perlfork Perl fork() information
perlnumber Perl number semantics

perlthrtut Perl threads tutorial

perlport Perl portability guide
perllocale Perl locale support
perluniintro Perl Unicode introduction
perlunicode Perl Unicode support
perlunifaq Perl Unicode FAQ
perluniprops Index of Unicode properties in Perl
perlunitut Perl Unicode tutorial
perlebcdic Considerations for running Perl on EBCDIC platforms

perlsec Perl security

perlmod Perl modules: how they work
perlmodlib Perl modules: how to write and use

`perlmodstyle` Perl modules: how to write modules with style
`perlmodinstall` Perl modules: how to install from CPAN
`perlnewmod` Perl modules: preparing a new module for distribution
`perlpragma` Perl modules: writing a user pragma

`perlutil` utilities packaged with the Perl distribution

`perlfiler` Perl source filters

`perldtrace` Perl's support for DTrace

`perlglossary` Perl Glossary

Internals and C Language Interface

`perlembed` Perl ways to embed perl in your C or C++ application
`perldebug` Perl debugging guts and tips
`perlxs` Perl XS application programming interface
`perlxsstut` Perl XS tutorial
`perlxs` Perl XS application programming interface
`perlxsstut` Perl XS C/Perl type conversion tools
`perlclib` Internal replacements for standard C library functions
`perlgu` Perl internal functions for those doing extensions
`perlcall` Perl calling conventions from C
`perlroapi` Perl method resolution plugin interface
`perlreapi` Perl regular expression plugin interface
`perlreguts` Perl regular expression engine internals

`perlapi` Perl API listing (autogenerated)
`perlintern` Perl internal functions (autogenerated)
`perliol` C API for Perl's implementation of IO in Layers
`perlpio` Perl internal IO abstraction interface

`perlhack` Perl hackers guide
`perlsource` Guide to the Perl source tree
`perlinterp` Overview of the Perl interpreter source and how it works
`perlhacktut` Walk through the creation of a simple C code patch
`perlhacktips` Tips for Perl core C code hacking
`perlpolicy` Perl development policies
`perlgit` Using git with the Perl repository

Miscellaneous

`perlbook` Perl book information
`perlcommunity` Perl community information

`perldoc` Look up Perl documentation in Pod format

`perlhists` Perl history records
`perldelta` Perl changes since previous version
`perl5180delta` Perl changes in version 5.18.0
`perl5161delta` Perl changes in version 5.16.1
`perl5162delta` Perl changes in version 5.16.2
`perl5163delta` Perl changes in version 5.16.3
`perl5160delta` Perl changes in version 5.16.0

perl5144delta Perl changes in version 5.14.4
perl5143delta Perl changes in version 5.14.3
perl5142delta Perl changes in version 5.14.2
perl5141delta Perl changes in version 5.14.1
perl5140delta Perl changes in version 5.14.0
perl5125delta Perl changes in version 5.12.5
perl5124delta Perl changes in version 5.12.4
perl5123delta Perl changes in version 5.12.3
perl5122delta Perl changes in version 5.12.2
perl5121delta Perl changes in version 5.12.1
perl5120delta Perl changes in version 5.12.0
perl5101delta Perl changes in version 5.10.1
perl5100delta Perl changes in version 5.10.0
perl589delta Perl changes in version 5.8.9
perl588delta Perl changes in version 5.8.8
perl587delta Perl changes in version 5.8.7
perl586delta Perl changes in version 5.8.6
perl585delta Perl changes in version 5.8.5
perl584delta Perl changes in version 5.8.4
perl583delta Perl changes in version 5.8.3
perl582delta Perl changes in version 5.8.2
perl581delta Perl changes in version 5.8.1
perl58delta Perl changes in version 5.8.0
perl561delta Perl changes in version 5.6.1
perl56delta Perl changes in version 5.6
perl5005delta Perl changes in version 5.005
perl5004delta Perl changes in version 5.004

perlexperiment A listing of experimental features in Perl

perlartistic Perl Artistic License
perlgpl GNU General Public License

Language-Specific

perlcn Perl for Simplified Chinese (in EUC-CN)
perljp Perl for Japanese (in EUC-JP)
perlko Perl for Korean (in EUC-KR)
perltw Perl for Traditional Chinese (in Big5)

Platform-Specific

perlaix Perl notes for AIX
perlamiga Perl notes for AmigaOS
perlbs2000 Perl notes for POSIX-BC BS2000
perlce Perl notes for WinCE
perlcygwin Perl notes for Cygwin
perldgux Perl notes for DG/UX
perldos Perl notes for DOS
perlfreesbsd Perl notes for FreeBSD
perlhaiku Perl notes for Haiku
perlhpx Perl notes for HP-UX
perlhurd Perl notes for Hurd
perlirix Perl notes for Irix
perllinux Perl notes for Linux
perlmacos Perl notes for Mac OS (Classic)

```
perlmacosx  Perl notes for Mac OS X
perlnetware Perl notes for NetWare
perlopenbsd Perl notes for OpenBSD
perlos2     Perl notes for OS/2
perlos390   Perl notes for OS/390
perlos400   Perl notes for OS/400
perlplan9   Perl notes for Plan 9
perlqnx     Perl notes for QNX
perlrisco   Perl notes for RISC OS
perlsolaris Perl notes for Solaris
perlsymbian Perl notes for Symbian
perltru64   Perl notes for Tru64
perlvms     Perl notes for VMS
perlvos     Perl notes for Stratus VOS
perlwin32   Perl notes for Windows
```

Stubs for Deleted Documents

```
perlboot
perlbot
perltodo
perltooc
perltoot
```

On a Unix-like system, these documentation files will usually also be available as manpages for use with the *man* program.

Some documentation is not available as man pages, so if a cross-reference is not found by *man*, try it with *perldoc*. *Perldoc* can also take you directly to documentation for functions (with the **-f** switch). See `perldoc --help` (or `perldoc perldoc` or `man perldoc`) for other helpful options *perldoc* has to offer.

In general, if something strange has gone wrong with your program and you're not sure where you should look for help, try making your code comply with **use strict** and **use warnings**. These will often point out exactly where the trouble is.

DESCRIPTION

Perl officially stands for Practical Extraction and Report Language, except when it doesn't.

Perl was originally a language optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It quickly became a good language for many system management tasks. Over the years, Perl has grown into a general-purpose programming language. It's widely used for everything from quick "one-liners" to full-scale application development.

The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal). It combines (in the author's opinion, anyway) some of the best features of **sed**, **awk**, and **sh**, making it familiar and easy to use for Unix users to whip up quick solutions to annoying problems. Its general-purpose programming facilities support procedural, functional, and object-oriented programming paradigms, making Perl a comfortable language for the long haul on major projects, whatever your bent.

Perl's roots in text processing haven't been forgotten over the years. It still boasts some of the most powerful regular expressions to be found anywhere, and its support for Unicode text is world-class. It handles all kinds of structured text, too, through an extensive collection of extensions. Those libraries, collected in the CPAN, provide ready-made solutions to an astounding array of problems. When they haven't set the standard themselves, they steal from the best -- just like Perl itself.

AVAILABILITY

Perl is available for most operating systems, including virtually all Unix-like platforms. See "*Supported Platforms*" in *perlport* for a listing.

ENVIRONMENT

See *perlrun*.

AUTHOR

Larry Wall <larry@wall.org>, with the help of oodles of other folks.

If your Perl success stories and testimonials may be of help to others who wish to advocate the use of Perl in their applications, or if you wish to simply express your gratitude to Larry and the Perl developers, please write to perl-thanks@perl.org.

FILES

`"@INC"` locations of perl libraries

SEE ALSO

http://www.perl.org/	the Perl homepage
http://www.perl.com/	Perl articles (O'Reilly)
http://www.cpan.org/	the Comprehensive Perl Archive
http://www.pm.org/	the Perl Mongers

DIAGNOSTICS

Using the `use strict` pragma ensures that all variables are properly declared and prevents other misuses of legacy Perl features.

The `use warnings` pragma produces some lovely diagnostics. One can also use the `-w` flag, but its use is normally discouraged, because it gets applied to all executed Perl code, including that not under your control.

See *perldiag* for explanations of all Perl's diagnostics. The `use diagnostics` pragma automatically turns Perl's normally terse warnings and errors into these longer forms.

Compilation errors will tell you the line number of the error, with an indication of the next token or token type that was to be examined. (In a script passed to Perl via `-e` switches, each `-e` is counted as one line.)

Setuid scripts have additional constraints that can produce error messages such as "Insecure dependency". See *perlsec*.

Did we mention that you should definitely consider using the **use warnings** pragma?

BUGS

The behavior implied by the **use warnings** pragma is not mandatory.

Perl is at the mercy of your machine's definitions of various operations such as type casting, `atof()`, and floating-point output with `sprintf()`.

If your stdio requires a seek or eof between reads and writes on a particular stream, so does Perl. (This doesn't apply to `sysread()` and `syswrite()`.)

While none of the built-in data types have any arbitrary size limits (apart from memory size), there are still a few arbitrary limits: a given variable name may not be longer than 251 characters. Line numbers displayed by diagnostics are internally stored as short integers, so they are limited to a maximum of 65535 (higher numbers usually being affected by wraparound).

You may mail your bug reports (be sure to include full configuration information as output by the

myconfig program in the perl source tree, or by `perl -v` to `perlbug@perl.org`. If you've succeeded in compiling perl, the *perlbug* script in the *utils/* subdirectory can be used to help mail in a bug report.

Perl actually stands for Pathologically Eclectic Rubbish Lister, but don't tell anyone I said that.

NOTES

The Perl motto is "There's more than one way to do it." Divining how many more is left as an exercise to the reader.

The three principal virtues of a programmer are Laziness, Impatience, and Hubris. See the Camel Book for why.