

NAME

perl – The Perl 5 language interpreter

SYNOPSIS

```
perl    [-sTtuUWX]          [-hv][-V[:configvar]]          [-cw][-d[t][:debugger]][-D[number/list]]
        [-pna][-Fpattern][-I[octal]][-O[octal/hexadecimal]]
        [-Idir][-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.

Overview

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

Tutorials

<code>perlreftut</code>	Perl references short introduction
<code>perldsc</code>	Perl data structures intro
<code>perllo1</code>	Perl data structures: arrays of arrays
<code>perlrequick</code>	Perl regular expressions quick start
<code>perlretut</code>	Perl regular expressions tutorial
<code>perloutut</code>	Perl OO tutorial for beginners
<code>perlperf</code>	Perl Performance and Optimization Techniques
<code>perlstyle</code>	Perl style guide
<code>perlcheat</code>	Perl cheat sheet
<code>perltrap</code>	Perl traps for the unwary
<code>perldebtut</code>	Perl debugging tutorial
<code>perlfaq</code>	Perl frequently asked questions
<code>perlfaq1</code>	General Questions About Perl
<code>perlfaq2</code>	Obtaining and Learning about Perl
<code>perlfaq3</code>	Programming Tools
<code>perlfaq4</code>	Data Manipulation
<code>perlfaq5</code>	Files and Formats
<code>perlfaq6</code>	Regexes
<code>perlfaq7</code>	Perl Language Issues
<code>perlfaq8</code>	System Interaction
<code>perlfaq9</code>	Networking

Reference Manual

<code>perlsyn</code>	Perl syntax
<code>perldata</code>	Perl data structures
<code>perlop</code>	Perl operators and precedence
<code>perlsub</code>	Perl subroutines
<code>perlfunc</code>	Perl built-in functions
<code>perlopentut</code>	Perl <code>open()</code> tutorial
<code>perlpacktut</code>	Perl <code>pack()</code> and <code>unpack()</code> tutorial
<code>perlpod</code>	Perl plain old documentation
<code>perlpodspec</code>	Perl plain old documentation format specification
<code>perlpodstyle</code>	Perl POD style guide
<code>perldiag</code>	Perl diagnostic messages
<code>perldeprecation</code>	Perl deprecations
<code>perllexwarn</code>	Perl warnings and their control
<code>perldebug</code>	Perl debugging
<code>perlvar</code>	Perl predefined variables
<code>perlre</code>	Perl regular expressions, the rest of the story
<code>perlrebackslash</code>	Perl regular expression backslash sequences
<code>perlrecharclass</code>	Perl regular expression character classes
<code>perlref</code>	Perl regular expressions quick reference
<code>perlref</code>	Perl references, the rest of the story
<code>perlform</code>	Perl formats
<code>perlobj</code>	Perl objects
<code>perltie</code>	Perl objects hidden behind simple variables
<code>perldbmfilter</code>	Perl DBM filters
<code>perlipc</code>	Perl interprocess communication
<code>perlfork</code>	Perl <code>fork()</code> information
<code>perlnumber</code>	Perl number semantics
<code>perlthrtut</code>	Perl threads tutorial
<code>perlport</code>	Perl portability guide
<code>perllocale</code>	Perl locale support
<code>perluniintro</code>	Perl Unicode introduction
<code>perlunicode</code>	Perl Unicode support
<code>perlunicook</code>	Perl Unicode cookbook
<code>perlunifaq</code>	Perl Unicode FAQ
<code>perluniprops</code>	Index of Unicode properties in Perl
<code>perlunitut</code>	Perl Unicode tutorial
<code>perlebcdic</code>	Considerations for running Perl on EBCDIC platforms
<code>perlsec</code>	Perl security
<code>perlmod</code>	Perl modules: how they work
<code>perlmodlib</code>	Perl modules: how to write and use
<code>perlmodstyle</code>	Perl modules: how to write modules with style
<code>perlmodinstall</code>	Perl modules: how to install from CPAN
<code>perlnewmod</code>	Perl modules: preparing a new module for distribution
<code>perlpragma</code>	Perl modules: writing a user pragma
<code>perlutil</code>	utilities packaged with the Perl distribution
<code>perlfilter</code>	Perl source filters

<code>perldtrace</code>	Perl's support for DTrace
<code>perlglossary</code>	Perl Glossary
Internals and C Language Interface	
<code>perlembed</code>	Perl ways to embed perl in your C or C++ application
<code>perldebbugs</code>	Perl debugging guts and tips
<code>perlxsut</code>	Perl XS tutorial
<code>perlxs</code>	Perl XS application programming interface
<code>perlxsstypemap</code>	Perl XS C/Perl type conversion tools
<code>perlclib</code>	Internal replacements for standard C library functions
<code>perlguts</code>	Perl internal functions for those doing extensions
<code>perlcall</code>	Perl calling conventions from C
<code>perlroapi</code>	Perl method resolution plugin interface
<code>perlreapi</code>	Perl regular expression plugin interface
<code>perlreguts</code>	Perl regular expression engine internals
<code>perlapi</code>	Perl API listing (autogenerated)
<code>perlintern</code>	Perl internal functions (autogenerated)
<code>perliol</code>	C API for Perl's implementation of IO in Layers
<code>perlpio</code>	Perl internal IO abstraction interface
<code>perlhack</code>	Perl hackers guide
<code>perlsource</code>	Guide to the Perl source tree
<code>perlinterp</code>	Overview of the Perl interpreter source and how it works
<code>perlhacktut</code>	Walk through the creation of a simple C code patch
<code>perlhacktips</code>	Tips for Perl core C code hacking
<code>perlpolicy</code>	Perl development policies
<code>perlgit</code>	Using git with the Perl repository
Miscellaneous	
<code>perlbook</code>	Perl book information
<code>perlcommunity</code>	Perl community information
<code>perldoc</code>	Look up Perl documentation in Pod format
<code>perlhists</code>	Perl history records
<code>perldelta</code>	Perl changes since previous version
<code>perl5280delta</code>	Perl changes in version 5.28.0
<code>perl5262delta</code>	Perl changes in version 5.26.2
<code>perl5261delta</code>	Perl changes in version 5.26.1
<code>perl5260delta</code>	Perl changes in version 5.26.0
<code>perl5244delta</code>	Perl changes in version 5.24.4
<code>perl5243delta</code>	Perl changes in version 5.24.3
<code>perl5242delta</code>	Perl changes in version 5.24.2
<code>perl5241delta</code>	Perl changes in version 5.24.1
<code>perl5240delta</code>	Perl changes in version 5.24.0
<code>perl5224delta</code>	Perl changes in version 5.22.4
<code>perl5223delta</code>	Perl changes in version 5.22.3
<code>perl5222delta</code>	Perl changes in version 5.22.2
<code>perl5221delta</code>	Perl changes in version 5.22.1
<code>perl5220delta</code>	Perl changes in version 5.22.0
<code>perl5203delta</code>	Perl changes in version 5.20.3
<code>perl5202delta</code>	Perl changes in version 5.20.2
<code>perl5201delta</code>	Perl changes in version 5.20.1
<code>perl5200delta</code>	Perl changes in version 5.20.0

perl5184delta	Perl changes in version 5.18.4
perl5182delta	Perl changes in version 5.18.2
perl5181delta	Perl changes in version 5.18.1
perl5180delta	Perl changes in version 5.18.0
perl5163delta	Perl changes in version 5.16.3
perl5162delta	Perl changes in version 5.16.2
perl5161delta	Perl changes in version 5.16.1
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

<code>perlaix</code>	Perl notes for AIX
<code>perlamiga</code>	Perl notes for AmigaOS
<code>perlandroid</code>	Perl notes for Android
<code>perlbs2000</code>	Perl notes for POSIX-BC BS2000
<code>perlce</code>	Perl notes for WinCE
<code>perlcygwin</code>	Perl notes for Cygwin
<code>perldos</code>	Perl notes for DOS
<code>perlfreesbsd</code>	Perl notes for FreeBSD
<code>perlhaiku</code>	Perl notes for Haiku
<code>perlhpx</code>	Perl notes for HP-UX
<code>perlhurd</code>	Perl notes for Hurd
<code>perlirix</code>	Perl notes for Irix
<code>perllinux</code>	Perl notes for Linux
<code>perlmacos</code>	Perl notes for Mac OS (Classic)
<code>perlmacosx</code>	Perl notes for Mac OS X
<code>perlnetware</code>	Perl notes for NetWare
<code>perlopenbsd</code>	Perl notes for OpenBSD
<code>perlos2</code>	Perl notes for OS/2
<code>perlos390</code>	Perl notes for OS/390
<code>perlos400</code>	Perl notes for OS/400
<code>perlplan9</code>	Perl notes for Plan 9
<code>perlqnx</code>	Perl notes for QNX
<code>perlricos</code>	Perl notes for RISC OS
<code>perlsolaris</code>	Perl notes for Solaris
<code>perlsymbian</code>	Perl notes for Symbian
<code>perlsynology</code>	Perl notes for Synology
<code>perltru64</code>	Perl notes for Tru64
<code>perlvms</code>	Perl notes for VMS
<code>perlvos</code>	Perl notes for Stratus VOS
<code>perlwin32</code>	Perl notes for Windows

Stubs for Deleted Documents

```
perlboot
perlbot
perlrepository
perltodo
perltooc
perltoot
```

On Debian systems, you need to install the **perl-doc** package which contains the majority of the standard Perl documentation and the *perldoc* program.

Extensive additional documentation for Perl modules is available, both those distributed with Perl and third-party modules which are packaged or locally installed.

You should be able to view Perl's documentation with your **man** (1) program or **perldoc** (1).

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

“@INC” above is a reference to the built-in variable of the same name; see perlvar for more information.

SEE ALSO

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