

NAME

perl – Practical Extraction and Report Language

SYNOPSIS

```
perl [-sTuU] [-hv] [-V [: configvar]] [-cw] [-d [: debugger]] [-D [number/list]] [-pna] [-F pattern] [-I [octal]] [-O [octal]] [-I dir] [-m [-] module] [-M [-] 'module...'] [-P] [-S] [-x [dir]] [-i [extension]] [-e 'command'] [--] [programfile] [argument...]
```

DESCRIPTION

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

OVERVIEW

perl	Perl overview (this section)
perlintro	Perl introduction for beginners
perltoc	Perl documentation table of contents

TUTORIALS**Tutorials**

perlreftut	Perl references short introduction
perldsc	Perl data structures intro
perllol	Perl data structures: arrays of arrays
perlrequick	Perl regular expressions quick start
perlretut	Perl regular expressions tutorial
perlboot	Perl OO tutorial for beginners
perltoot	Perl OO tutorial, part 1
perltoc	Perl OO tutorial, part 2
perlbot	Perl OO tricks and examples
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
perlrun	Perl execution and options

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
perlref	Perl regular expressions quick reference
perlref	Perl references, the rest of the story
perlform	Perl formats
perlobj	Perl objects
perltie	Perl objects hidden behind simple variables
perldbfilter	Perl DBM filters
perlipc	Perl interprocess communication
perlfork	Perl fork() information
perlnumber	Perl number semantics
perlthrtut	Perl threads tutorial
perlothrtut	Old Perl threads tutorial
perlport	Perl portability guide
perllocale	Perl locale support
perluniintro	Perl Unicode introduction
perlunicode	Perl Unicode support
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
perlutil	utilities packaged with the Perl distribution
perlcompile	Perl compiler suite intro
perlfiler	Perl source filters

INTERNALS AND C LANGUAGE INTERFACE

perlembed	Perl ways to embed perl in your C or C++ application
perldebbugs	Perl debugging guts and tips
perlxsut	Perl XS tutorial
perlxs	Perl XS application programming interface
perlclib	Internal replacements for standard C library functions
perlguts	Perl internal functions for those doing extensions
perlcall	Perl calling conventions from C
perlapi	Perl API listing (autogenerated)
perlintern	Perl internal functions (autogenerated)
perliol	C API for Perl's implementation of IO in Layers
perlapiio	Perl internal IO abstraction interface
perlhack	Perl hackers guide

MISCELLANEOUS

perlbook	Perl book information
perltodo	Perl things to do

perldoc	Look up Perl documentation in Pod format
perlhists	Perl history records
perldelta	Perl changes since previous version
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
perl573delta	Perl changes in version 5.7.3
perl572delta	Perl changes in version 5.7.2
perl571delta	Perl changes in version 5.7.1
perl570delta	Perl changes in version 5.7.0
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
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

perlsolaris	Perl notes for Solaris
-------------	------------------------

Platform-Specific

If you're new to Perl, you should start with **perlintro**, which is a general intro for beginners and provides some background to help you navigate the rest of Perl's extensive documentation. For ease of access, the Perl manual has been split up into several sections.

The manpages listed above are installed in the **/usr/perl5/man/** directory.

Extensive additional documentation for Perl modules is available. This additional documentation is in the **/usr/perl5/man** directory. Some of this additional documentation is distributed standard with Perl, but you'll also find documentation for any customer-installed third-party modules there.

You can view Perl's documentation with **man(1)** by including **/usr/perl5/man** in the **MANPATH** environment variable. Notice that running **catman(1M)** on the Perl manual pages is not supported. For other Solaris-specific details, see the **NOTES** section below.

You can also use the supplied **/usr/perl5/bin/perldoc** script to view Perl information.

If something strange has gone wrong with your program and you're not sure where you should look for help, try the **-w** switch first. It will often point out exactly where the trouble is.

Perl is a language optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It's also a good language for many system management tasks. The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal).

Perl combines (in the author's opinion, anyway) some of the best features of C, **sed**, **awk**, and **sh**, so people familiar with those languages should have little difficulty with it. (Language historians will also note some vestiges of **csh**, Pascal, and even BASIC-PLUS.) Expression syntax corresponds closely to C expression syntax. Unlike most Unix utilities, Perl does not arbitrarily limit the size of your data -if you've got the memory, Perl can slurp in your whole file as a single string. Recursion is of unlimited depth. And the tables used by hashes (sometimes called "associative arrays") grow as necessary to

prevent degraded performance. Perl can use sophisticated pattern matching techniques to scan large amounts of data quickly. Although optimized for scanning text, Perl can also deal with binary data, and can make dbm files look like hashes. Setuid Perl scripts are safer than C programs through a dataflow tracing mechanism that prevents many stupid security holes.

If you have a problem that would ordinarily use **sed** or **awk** or **sh**, but it exceeds their capabilities or must run a little faster, and you don't want to write the silly thing in C, then Perl may be for you. There are also translators to turn your **sed** and **awk** scripts into Perl scripts.

But wait, there's more...

Begun in 1993 (see **perlhist**), Perl version 5 is nearly a complete rewrite that provides the following additional benefits:

- Modularity and reusability using innumerable modules Described in **perlmod**, **perlmodlib**, and **perlmodinstall**.
- Embeddable and extensible Described in **perlembed**, **perlxsut**, **perlxs**, **perlcalls**, **perlguts**, and **xsubpp**.
- Roll-your-own magic variables (including multiple simultaneous DBM implementations). Described in **perltie** and **AnyDBM_File**.
- Subroutines can now be overridden, autoloaded, and prototyped. Described in **perlsub**.
- Arbitrarily nested data structures and anonymous functions. Described in **perlreftut**, **perlref**, **perldsc**, and **perllol**.
- Object-oriented programming. Described in **perlobj**, **perlboot**, **perltoot**, **perltoc**, and **perlbot**.
- Support for light-weight processes (threads). Described in **perlthrtut** and **threads**.
- Support for Unicode, internationalization, and localization Described in **perluniintro**, **perllocale** and **Locale::Maketext**.
- Lexical scoping. Described in **perlsub**.
- Regular expression enhancements. Described in **perlre**, with additional examples in **perllop**.
- Enhanced debugger and interactive Perl environment, with integrated editor support. Described in **perldebtut**, **perldebug** and **perldebguts**.
- POSIX 1003.1 compliant library Described in POSIX.

Okay, that's *definitely* enough hype.

ENVIRONMENT VARIABLES

The Perl shipped with Solaris is installed under **/usr/perl5** rather than the default **/usr/local** location. This is so that it can coexist with a customer-installed Perl in the default **/usr/local** location.

Any additional modules that you choose to install will be placed in the **/usr/perl5/site_perl/5.8.4** directory. The **/usr/perl5/vendor_perl** directory is reserved for SMI-provided modules.

Notice that the Perl utility scripts such as **perldoc** and **perlbug** are in the **/usr/perl5/bin** directory, so if you wish to use them you need to include **/usr/perl5/bin** in your PATH environment variable.

See also the **perlrun** mapage.

AUTHOR

Larry Wall, 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

ATTRIBUTES

See **attributes(5)** for descriptions of the following attributes:

tab() allbox; cw(2.750000i)| cw(2.750000i) lw(2.750000i)| lw(2.750000i). ATTRIBUTE
TYPEATTRIBUTE VALUE AvailabilityT{ SUNWperl584core, SUNWperl584usr, SUNWperl584man,
SUNWpl5u, SUNWpl5v SUNWpl5p, SUNWpl5m

See below. T} Interface StabilitySee below.

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

The Script interface is Evolving. The XSUB interface is Evolving. The Binary interface is Unstable. The Directory layout is Evolving.

SEE ALSO

a2p **awk** to **perl** translator

s2p **sed** to **perl** translator

http://www.perl.com Perl home page

http://www.perl.com/CPAN The Comprehensive Perl Archive

http://www.perl.org Perl Mongers (Perl user groups)

DIAGNOSTICS

The **'use warnings'** pragma (and the **-w** switch) produce some lovely diagnostics.

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 `-w` switch?

NOTES

Perl 5.8.4 has been built to be largefile-aware and to use 64-bit integers, although the interpreter itself is a 32-bit application (LP32). To view detailed configuration information, use `perl -V` and `perlbug -dv`.

If you wish to build and install add-on modules from CPAN using `gcc`, you can do so using the `/usr/perl5/5.8.4/bin/perlgcc` script - see `perlgcc(1)` for details.

If you wish to build and install your own version of Perl, you should NOT remove the 5.8.4 version of perl under `/usr/perl5`, as it is required by several system utilities. The Perl package names are as follows:

<code>SUNWperl584core</code>	Perl 5.8.4 (Core files)
<code>SUNWperl584usr</code>	Perl 5.8.4 (Non-core files)
<code>SUNWperl584man</code>	Perl 5.8.4 (Manual pages)

Solaris 10 also ships with the 5.6.1 version of Perl that was included in Solaris 9. If you are upgrading your system and wish to continue to use Perl 5.6.1 as the default Perl version you should refer to the `perlsolaris` manpage for details of how to do this. Note that you should upgrade your installation to use Perl 5.8.4 as soon as is practicable, as Perl 5.6.1 may be removed in a future release.

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.

BUGS

The `-w` switch 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.