NAME
perl – Practical Extraction and Report Language

SYNOPSIS
perl [-sTuU] [-hv] [-V [ : configvar]] [-cw] [-d [ : debugger]] [-D [number/list]] [-pna] [-F pattern] [-I [octal]] [-0 [octal]] [-i dir] [-m [-] module] [-M [-] 'module...' [-P] [-S] [-x [dir]] [-i [extension]] [-e 'command'] [-t] [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
perltreftut Perl references short introduction
perldisc Perl data structures intro
perllol Perl data structures: arrays of arrays
perlfrequick Perl regular expressions quick start
perlreftut Perl regular expressions tutorial
perlboot Perl OO tutorial for beginners
perltoot Perl OO tutorial, part 1
perltooc 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
perldebut Perl debugging tutorial
perlfqaq Perl frequently asked questions
perlfqaq1 General Questions About Perl
perlfqaq2 Obtaining and Learning about Perl
perlfqaq3 Programming Tools
perlfqaq4 Data Manipulation
perlfqaq5 Files and Formats
perlfqaq6 Regexes
perlfqaq7 Perl Language Issues
perlfqaq8 System Interaction
perlfqaq9 Networking

REFERENCE MANUAL
perlsyn Perl syntax
perldata Perl data structures
perllop Perl operators and precedence
perls sub Perl subroutines
perlfunc Perl built-in functions
perlreftut Perl open() tutorial
perlpacktut Perl pack() and unpack() tutorial
perlpod Perl plain old documentation
perlpo dspec Perl plain old documentation format specification
perlrunc Perl execution and options

SunOS 5.10 Last change: 30 Jul 2004
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>perldiag</td>
<td>Perl diagnostic messages</td>
</tr>
<tr>
<td>perlexwarn</td>
<td>Perl warnings and their control</td>
</tr>
<tr>
<td>perldebug</td>
<td>Perl debugging</td>
</tr>
<tr>
<td>perlvar</td>
<td>Perl predefined variables</td>
</tr>
<tr>
<td>perlre</td>
<td>Perl regular expressions, the rest of the story</td>
</tr>
<tr>
<td>perlreref</td>
<td>Perl regular expressions quick reference</td>
</tr>
<tr>
<td>perlref</td>
<td>Perl references, the rest of the story</td>
</tr>
<tr>
<td>perlfour</td>
<td>Perl formats</td>
</tr>
<tr>
<td>perlobj</td>
<td>Perl objects</td>
</tr>
<tr>
<td>perltie</td>
<td>Perl objects hidden behind simple variables</td>
</tr>
<tr>
<td>perldbmfilter</td>
<td>Perl DBM filters</td>
</tr>
<tr>
<td>perlpc</td>
<td>Perl interprocess communication</td>
</tr>
<tr>
<td>perlfork</td>
<td>Perl fork() information</td>
</tr>
<tr>
<td>perlnumber</td>
<td>Perl number semantics</td>
</tr>
<tr>
<td>perlthrtut</td>
<td>Perl threads tutorial</td>
</tr>
<tr>
<td>perlothrtut</td>
<td>Old Perl threads tutorial</td>
</tr>
<tr>
<td>perlport</td>
<td>Perl portability guide</td>
</tr>
<tr>
<td>perllocale</td>
<td>Perl locale support</td>
</tr>
<tr>
<td>perluniintro</td>
<td>Perl Unicode introduction</td>
</tr>
<tr>
<td>perlunicode</td>
<td>Perl Unicode support</td>
</tr>
<tr>
<td>perlebcdic</td>
<td>Considerations for running Perl on EBCDIC platforms</td>
</tr>
<tr>
<td>perlsec</td>
<td>Perl security</td>
</tr>
<tr>
<td>perlmmod</td>
<td>Perl modules: how they work</td>
</tr>
<tr>
<td>perlmodlib</td>
<td>Perl modules: how to write and use</td>
</tr>
<tr>
<td>perlmodstyle</td>
<td>Perl modules: how to write modules with style</td>
</tr>
<tr>
<td>perlmodinstall</td>
<td>Perl modules: how to install from CPAN</td>
</tr>
<tr>
<td>perlnewmod</td>
<td>Perl modules: preparing a new module for distribution</td>
</tr>
<tr>
<td>perlutil</td>
<td>utilities packaged with the Perl distribution</td>
</tr>
<tr>
<td>perlcflow</td>
<td>Perl compiler suite intro</td>
</tr>
<tr>
<td>perlfilter</td>
<td>Perl source filters</td>
</tr>
<tr>
<td>perliol</td>
<td>C API for Perl's implementation of IO in Layers</td>
</tr>
<tr>
<td>perlapi</td>
<td>Perl API listing (autogenerated)</td>
</tr>
<tr>
<td>perlintern</td>
<td>Perl internal functions (autogenerated)</td>
</tr>
<tr>
<td>perlclib</td>
<td>Internal replacements for standard C library functions</td>
</tr>
<tr>
<td>perlcall</td>
<td>Perl calling conventions from C</td>
</tr>
<tr>
<td>perlguts</td>
<td>Perl internal functions for those doing extensions</td>
</tr>
<tr>
<td>perlbian</td>
<td>Perl hackers guide</td>
</tr>
<tr>
<td>perlbook</td>
<td>Perl book information</td>
</tr>
<tr>
<td>perltodo</td>
<td>Perl things to do</td>
</tr>
</tbody>
</table>
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, perlxstut, perlxst, perlcall, 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 perlextut, perlref, perloldc, and perllol.
- Object-oriented programming. Described in perlobj, perlboot, perltoot, perltooc, and perlbot.
- Support for light-weight processes (threads). Described in perlthrtut and threads.
- Support for Unicode, internationalization, and localization Described in perluniintro, perllo-
cale and Locale::Maketext.
- Lexical scoping. Described in perlsub.
- Regular expression enhancements. Described in perlre, with additional examples in perlop.
- Enhanced debugger and interactive Perl environment, with integrated editor support. Described in perldebtut, perlddebug and perldebuguts.
- 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 `perlrn` manpage.

**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)`
- `lw(2.750000i)`
- `cw(2.750000i)`
- `lw(2.750000i)`

ATTRIBUTE TYPE ATTRIBUTE VALUE

Availability

SunWperl584core, SunWperl584usr, SunWperl584man, SunWpl5u, SunWpl5v, SunWpl5p, SunWpl5m

See below.

Interface Stability

See 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 diagnosticts` 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 depen-
dency". See perlsec.

Did we mention that you should definitely consider using the \texttt{-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
\texttt{/usr/perl5/5.8.4/bin/perlgcc} script - see \texttt{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 \texttt{/usr/perl5}, as it is required by several system utilities. The Perl package names are as fol-

\begin{verbatim}
SUNWperl584core Perl 5.8.4 (Core files)
SUNWperl584usr Perl 5.8.4 (Non-core files)
SUNWperl584man Perl 5.8.4 (Manual pages)
\end{verbatim}

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 \texttt{-w} switch is not mandatory.

Perl is at the mercy of your machine’s definitions of various operations such as type casting, \texttt{atof()}, and
floating-point output with \texttt{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 \texttt{sysread()} and \texttt{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 \texttt{perl -V}) to \texttt{perlbug@perl.org}. If you’ve succeeded
in compiling perl, the \texttt{perlbug} script in the \texttt{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.