

B<This program> will read the given input file(s) and do something useful with the contents thereof.

=cut

See *Pod::Usage* for details.

A bug that prevented the non-option call-back <> from being specified as the first argument has been fixed.

To specify the characters < and > as option starters, use ><. Note, however, that changing option starters is strongly deprecated.

IO

write() and syswrite() will now accept a single-argument form of the call, for consistency with Perl's syswrite().

You can now create a TCP-based IO::Socket::INET without forcing a connect attempt. This allows you to configure its options (like making it non-blocking) and then call connect() manually.

A bug that prevented the IO::Socket::protocol() accessor from ever returning the correct value has been corrected.

IO::Socket::connect now uses non-blocking IO instead of alarm() to do connect timeouts.

IO::Socket::accept now uses select() instead of alarm() for doing timeouts.

IO::Socket::INET->new now sets \$! correctly on failure. \$@ is still set for backwards compatibility.

JPL

Java Perl Lingo is now distributed with Perl. See jpl/README for more information.

lib

use lib now weeds out any trailing duplicate entries. no lib removes all named entries.

Math::BigInt

The bitwise operations <<, >>, &, |, and ~ are now supported on bigints.

Math::Complex

The accessor methods Re, Im, arg, abs, rho, and theta can now also act as mutators (accessor \$z->Re(), mutator \$z->Re(3)).

The class method display_format and the corresponding object method display_format, in addition to accepting just one argument, now can also accept a parameter hash. Recognized keys of a parameter hash are "style", which corresponds to the old one parameter case, and two new parameters: "format", which is a printf()-style format string (defaults usually to "%.15g", you can revert to the default by setting the format string to undef) used for both parts of a complex number, and "polar_pretty_print" (defaults to true), which controls whether an attempt is made to try to recognize small multiples and rationals of pi (2pi, pi/2) at the argument (angle) of a polar complex number.

The potentially disruptive change is that in list context both methods now *return the parameter hash*, instead of only the value of the "style" parameter.

Math::Trig

A little bit of radial trigonometry (cylindrical and spherical), radial coordinate conversions, and the great circle distance were added.

Pod::Parser, Pod::InputObjects

Pod::Parser is a base class for parsing and selecting sections of pod documentation from an input stream. This module takes care of identifying pod paragraphs and commands in the input and hands off the parsed paragraphs and commands to user-defined methods which are free to interpret or translate them as they see fit.

Pod::InputObjects defines some input objects needed by Pod::Parser, and for advanced users of Pod::Parser that need more about a command besides its name and text.

As of release 5.6.0 of Perl, Pod::Parser is now the officially sanctioned "base parser code" recommended for use by all pod2xxx translators. Pod::Text (pod2text) and Pod::Man (pod2man) have already been converted to use