

NAME

perl5115delta - what is new for perl v5.11.5

DESCRIPTION

This document describes differences between the 5.11.4 release and the 5.11.5 release.

If you are upgrading from an earlier release such as 5.11.3, first read *perl5114delta*, which describes differences between 5.11.3 and 5.11.4.

Core Enhancements

32-bit limit on substr arguments removed

The 32-bit limit on `substr` arguments has now been removed. The full range of the system's signed and unsigned integers is now available for the `pos` and `len` arguments.

Modules and Pragmata

Pragmata Changes

`version`

Upgraded from version 0.81 to 0.82.

The `is_lax` and `is_strict` functions can now be optionally exported to the caller's namespace and are also now documented.

Undefined version objects are now uninitialized with zero rather than `undef`.

Updated Modules

`B::Debug`

Upgraded from version 1.11 to 1.12.

`CPAN`

Upgraded from version 1.94_53 to 1.94_56.

This resolves RT #72362, in which CPAN was ignoring `configure_requires`, and RT #72348, in which the command `o conf init` in the CPAN shell could cause an exception to be thrown.

This module is also now built in a less specialized way, which resolves a problem that caused `make after make clean` to fail, fixing RT #72218.

`CPANPLUS::Dist::Build`

Upgraded from version 0.44 to 0.46.

This makes the `prereq` resolving fall back to `_build/` querying if the `prereq_data` action fails.

`Pod::Perldoc`

Upgraded from version 3.15_01 to 3.15_02.

`Pod::Plainer`

Upgraded from version 1.01 to 1.02.

`Safe`

Upgraded from version 2.21 to 2.22.

This resolves RT #72700, in which an exception thrown from a closure was getting lost.

`Socket`

Upgraded from version 1.85 to 1.86.

This makes the new `Socket` implementation of `inet_pton` consistent with the existing `Socket6` implementation of `inet_pton`, fixing RT #72884.

podlators

Upgraded from version 2.2.2 to 2.3.1.

Changes to Existing Documentation

The syntax `unless (EXPR) BLOCK else BLOCK` is now documented as valid, as is the syntax `unless (EXPR) BLOCK elsif (EXPR) BLOCK ... else BLOCK`, although actually using the latter may not be the best idea for the readability of your source code.

Installation and Configuration Improvements

Configuration improvements

Support for SystemTap's `dtrace` compatibility layer has been added and an issue with linking `miniperl` has been fixed in the process.

`less -R` is now used instead of `less` for `groff`'s new usage of ANSI escape codes by setting `$Config{less}` (and thereby `$Config{pager}`), which fixes RT #72156.

`USE_PERL_ATOF` is now reported in the compile-time options listed by the `-v` switch.

Selected Bug Fixes

- Arbitrary whitespace is now allowed between `NAME` and `VERSION` in package `NAME VERSION;` statements. (Fixes RT #72432)
- A panic caused by trying to load `chardnames` when the parser is already in error (e.g. by a missing `my` under `use strict;`) is now averted. This was a regression since Perl 5.10.0. (Fixes RT #72590)
- Reading `$!` no longer causes a SEGV for out of range `errno` values. (Fixes RT #72850)
- A possible SEGV in `/\N{...}/` has been fixed. This was a regression since Perl 5.10.
- A possible SEGV when freeing a scalar that was upgraded to an `SVt_REGEXP` type from a `simple(r)` scalar has been fixed.
- A type conversion bug in `gmtime64` that caused it to break around `2**48` has been fixed.
- Interpolating a regex that makes use of the `chardnames` pragma will no longer cause a run-time error. (Fixes RT #56444)
- Array references assigned to `*FOO::ISA` now have the necessary magic added to them to catch any further updates to the new `@ISA`. (Fixes RT #72866)
- Filehandles are now always blessed into `IO::File`, which, together with some suitable manipulation of `@IO::File::ISA`, fixes a breakage introduced in Perl 5.11.3 by a change that always blessed filehandles into `IO::Handle` rather than checking for `FileHandle` first.
- A change in the behaviour of `warnings::enabled` and `warnings::warnif` in Perl 5.10.0 that wasn't documented at the time is now documented in *perl5100delta*. (Fixes RT #62522)
- RT #71504 is now fixed by simply skipping the tests that failed on OpenBSD with `ithreads` and `perlio`.

New or Changed Diagnostics

- The fatal error `Malformed UTF-8 returned by \N` is now produced if the `chardnames` handler returns malformed UTF-8.
- If an unresolved named character or sequence was encountered when compiling a regex pattern then the fatal error `\N{NAME} must be resolved by the lexer` is now produced. This can happen, for example, when using a single-quotish context like `$re = '\N{SPACE}'; $re;`. See *perldiag* for more examples of how the lexer can get bypassed.

- The fatal error `Invalid hexadecimal number in \\N{U+...}` will be produced if the character constant represented by `...` is not a valid hexadecimal number.
- The new meaning of `\N` as `[^\n]` is not valid in a bracketed character class, just like `.` in a character class loses its special meaning, and will cause the fatal error `\\N in a character class must be a named character: \\N{...}`.
- The rules on what is legal for the `...` in `\N{...}` have been tightened up so that unless the `...` begins with an alphabetic character and continues with a combination of alphanumerics, dashes, spaces, parentheses or colons then the warning `Deprecated character(s) in \\N{...}` starting at `'%s'` is now issued.
- The warning `Using just the first characters returned by \\N{}` will be issued if the `charnames` handler returns a sequence of characters which exceeds the limit of the number of characters that can be used. The message will indicate which characters were used and which were discarded.
- Currently, all but the first of the several characters that the `charnames` handler may return are discarded when used in a regular expression pattern bracketed character class. If this happens then the warning `Using just the first character returned by \\N{}` in character class will be issued.
- The warning `Missing right brace on \\N{}` or unescaped left brace after `\\N`. Assuming the latter will be issued if Perl encounters a `\N{` but doesn't find a matching `}`. In this case Perl doesn't know if it was mistakenly omitted, or if "match non-newline" followed by "match a {" was desired. It assumes the latter because that is actually a valid interpretation as written, unlike the other case. If you meant the former, you need to add the matching right brace. If you did mean the latter, you can silence this warning by writing instead `\N\{`.
- `gmtime` and `localtime` called with numbers smaller than they can reliably handle will now issue the warnings `gmtime(%.0f) too small` and `localtime(%.0f) too small`.

New Tests

t/op/filehandle.t

Tests some suitably portable filetest operators to check that they work as expected, particularly in the light of some internal changes made in how filehandles are blessed.

t/op/time_loop.t

Tests that times greater than 2^{*63} , which can now be handed to `gmtime` and `localtime`, do not cause an internal overflow or an excessively long loop.

Known Problems

Perl 5.11.5 is a development release leading up to Perl 5.12.0. Some notable known problems found in 5.11.5 are listed as dependencies of RT #69710, the Perl 5 version 12 meta-ticket.

Acknowledgements

Perl 5.11.5 represents approximately one month of development since Perl 5.11.4 and contains 9618 lines of changes across 151 files from 33 authors and committers:

Ævar Arnfjörð Bjarmason, Abigail, brian d foy, Chris Williams, David Golden, David Mitchell, Eric Brine, Frank Wiegand, Gisle Aas, H.Merijn Brand, Jan Dubois, Jesse Vincent, Jim Cromie, John Peacock, Josh ben Jore, Karl Williamson, Marcus Holland-Moritz, Michael G Schwern, Nicholas Clark, Offer Kaye, Philippe Bruhat (Book), Rafael Garcia-Suarez, Reini Urban, Ricardo Signes, Robin Barker, Slaven Rezić, Steffen Mueller, Steve Hay, Steve Peters, Tim Bunce, Todd Rinaldo, Tony Cook and Vincent Pit.

Many of the changes included in this version originated in the CPAN modules included in Perl's core.

We're grateful to the entire CPAN community for helping Perl to flourish.

Reporting Bugs

If you find what you think is a bug, you might check the articles recently posted to the `comp.lang.perl.misc` newsgroup and the perl bug database at <http://rt.perl.org/perlbug/> . There may also be information at <http://www.perl.org/> , the Perl Home Page.

If you believe you have an unreported bug, please run the **perlbug** program included with your release. Be sure to trim your bug down to a tiny but sufficient test case. Your bug report, along with the output of `perl -v`, will be sent off to `perlbug@perl.org` to be analyzed by the Perl porting team.

If the bug you are reporting has security implications, which make it inappropriate to send to a publicly archived mailing list, then please send it to `perl5-security-report@perl.org`. This points to a closed subscription unarchived mailing list, which includes all the core committers, who be able to help assess the impact of issues, figure out a resolution, and help co-ordinate the release of patches to mitigate or fix the problem across all platforms on which Perl is supported. Please only use this address for security issues in the Perl core, not for modules independently distributed on CPAN.

SEE ALSO

The *Changes* file for an explanation of how to view exhaustive details on what changed.

The *INSTALL* file for how to build Perl.

The *README* file for general stuff.

The *Artistic* and *Copying* files for copyright information.