NAME

perl5221delta - what is new for perl v5.22.1

DESCRIPTION

This document describes differences between the 5.22.0 release and the 5.22.1 release.

If you are upgrading from an earlier release such as 5.20.0, first read perl5220delta, which describes differences between 5.20.0 and 5.22.0.

Incompatible Changes

There are no changes intentionally incompatible with 5.20.0 other than the following single exception, which we deemed to be a sensible change to make in order to get the new "\b{wb}" and (in particular) "\b{sb}" features sane before people decided they're worthless because of bugs in their Perl 5.22.0 implementation and avoided them in the future. If any others exist, they are bugs, and we request that you submit a report. See "Reporting Bugs" below.

Bounds Checking Constructs

Several bugs, including a segmentation fault, have been fixed with the bounds checking constructs (introduced in Perl 5.22) "\b{gcb}", "\b{sb}", "\b{wb}", "\B{gcb}", "\B{sb}", and "\B{wb}". All the "\B{}" ones now match an empty string; none of the "\b{}" ones do. [GH #14976] <https://github.com/Perl/perl5/issues/14976>

Modules and Pragmata

Updated Modules and Pragmata

⊕   Module::CoreList has been upgraded from version 5.20150520 to 5.20151213.

⊕   PerlIO::scalar has been upgraded from version 0.22 to 0.23.

⊕   POSIX has been upgraded from version 1.53 to 1.53_01.

    If "POSIX::strerror" was passed $! as its argument then it accidentally cleared $!. This has been fixed. [GH #14951] <https://github.com/Perl/perl5/issues/14951>

⊕   Storable has been upgraded from version 2.53 to 2.53_01.

⊕   warnings has been upgraded from version 1.32 to 1.34.

    The "warnings::enabled" example now actually uses "warnings::enabled". [GH #14905] <https://github.com/Perl/perl5/issues/14905>

⊕   Win32 has been upgraded from version 0.51 to 0.52.

This has been updated for Windows 8.1, 10 and 2012 R2 Server.

Documentation
Changes to Existing Documentation
perltie

⊕   The usage of "FIRSTKEY" and "NEXTKEY" has been clarified.

perlvar

⊕   The specific true value of $!{E...} is now documented, noting that it is subject to change and not
    guaranteed.

Diagnostics
The following additions or changes have been made to diagnostic output, including warnings and fatal
error messages. For the complete list of diagnostic messages, see perldiag.

Changes to Existing Diagnostics
⊕   The "printf" and "sprintf" builtins are now more careful about the warnings they emit: argument
    reordering now disables the "redundant argument" warning in all cases. [GH #14772]
    <https://github.com/Perl/perl5/issues/14772>

Configuration and Compilation
⊕   Using the "NO_HASH_SEED" define in combination with the default hash algorithm
    "PERL_HASH_FUNC_ONE_AT_A_TIME_HARD" resulted in a fatal error while compiling the
    interpreter, since Perl 5.17.10. This has been fixed.

⊕   Configuring with ccflags containing quotes (e.g.
    "-Accflags='-DAPPLLIB_EXP=\"/usr/libperl\"'") was broken in Perl 5.22.0 but has now been
    fixed again. [GH #14732] <https://github.com/Perl/perl5/issues/14732>

Platform Support
Platform-Specific Notes
IRIX

⊕   Under some circumstances IRIX stdio **fgetc()** and **fread()** set the errno to "ENOENT", which
    made no sense according to either IRIX or POSIX docs. Errno is now cleared in such cases.
    [GH #14557] <https://github.com/Perl/perl5/issues/14557>

⊕ Problems when multiplying long doubles by infinity have been fixed. [GH #14993]
       <https://github.com/Perl/perl5/issues/14993>

⊕ All tests pass now on IRIX with the default build configuration.

Selected Bug Fixes

⊕ "qr/(?[()])/" no longer segfaults, giving a syntax error message instead. [GH #14851]
       <https://github.com/Perl/perl5/issues/14851>

⊕ Regular expression possessive quantifier Perl 5.20 regression now fixed.
       "qr/"*PAT*"{"*min,max*"}+""/" is supposed to behave identically to "qr/(?>"*PAT*"{"*min,max*"})/".
       Since Perl 5.20, this didn't work if *min* and *max* were equal. [GH #14857]
       <https://github.com/Perl/perl5/issues/14857>

⊕ Certain syntax errors in "Extended Bracketed Character Classes" in perlrecharclass caused panics
       instead of the proper error message. This has now been fixed. [GH #15016]
       <https://github.com/Perl/perl5/issues/15016>

⊕ "BEGIN <>" no longer segfaults and properly produces an error message. [GH #13546]
       <https://github.com/Perl/perl5/issues/13546>

⊕ A regression from Perl 5.20 has been fixed, in which some syntax errors in "(?[...])" constructs
       within regular expression patterns could cause a segfault instead of a proper error message. [GH
       #14933] <https://github.com/Perl/perl5/issues/14933>

⊕ Another problem with "(?[...])" constructs has been fixed wherein things like "\c]" could cause
       panics. [GH #14934] <https://github.com/Perl/perl5/issues/14934>

⊕ In Perl 5.22.0, the logic changed when parsing a numeric parameter to the -C option, such that the
       successfully parsed number was not saved as the option value if it parsed to the end of the
       argument. [GH #14748] <https://github.com/Perl/perl5/issues/14748>

⊕ Warning fatality is now ignored when rewinding the stack. This prevents infinite recursion when
       the now fatal error also causes rewinding of the stack. [GH #14319]
       <https://github.com/Perl/perl5/issues/14319>

⊕ A crash with "%::=(); J->${\"::"}" has been fixed. [GH #14790]
       <https://github.com/Perl/perl5/issues/14790>

⊕ Nested quantifiers such as "/.{1}??/" should cause perl to throw a fatal error, but were being

silently accepted since Perl 5.20.0. This has been fixed. [GH #14960]
<https://github.com/Perl/perl5/issues/14960>

⊕   Regular expression sequences such as "/(?i/" (and similarly with other recognized flags or combination of flags) should cause perl to throw a fatal error, but were being silently accepted since Perl 5.18.0. This has been fixed. [GH #14931]
<https://github.com/Perl/perl5/issues/14931>

⊕   A bug in hexadecimal floating point literal support meant that high-order bits could be lost in cases where mantissa overflow was caused by too many trailing zeros in the fractional part. This has been fixed. [GH #15032] <https://github.com/Perl/perl5/issues/15032>

⊕   Another hexadecimal floating point bug, causing low-order bits to be lost in cases where the last hexadecimal digit of the mantissa has bits straddling the limit of the number of bits allowed for the mantissa, has also been fixed. [GH #15033] <https://github.com/Perl/perl5/issues/15033>

⊕   Further hexadecimal floating point bugs have been fixed: In some circumstances, the %a format specifier could variously lose the sign of the negative zero, fail to display zeros after the radix point with the requested precision, or even lose the radix point after the leftmost hexadecimal digit completely.

⊕   A crash caused by incomplete expressions within "/(?[ ])/" (e.g. "/(?[[0]+()+])/") has been fixed. [GH #15045] <https://github.com/Perl/perl5/issues/15045>

Acknowledgements

The list above is almost certainly incomplete as it is automatically generated from version control history. In particular, it does not include the names of the (very much appreciated) contributors who reported issues to the Perl bug tracker.

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.

For a more complete list of all of Perl's historical contributors, please see the *AUTHORS* file in the Perl source distribution.

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 https://rt.perl.org/ . 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 analysed 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 will 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.