



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'perl5320delta.1'

\$ man perl5320delta.1

PERL5320DELTA(1) Perl Programmers Reference Guide PERL5320DELTA(1)

NAME

perl5320delta - what is new for perl v5.32.0

DESCRIPTION

This document describes differences between the 5.30.0 release and the 5.32.0 release.

If you are upgrading from an earlier release such as 5.28.0, first read perl5300delta, which describes differences between 5.28.0 and 5.30.0.

Core Enhancements

The isa Operator

A new experimental infix operator called "isa" tests whether a given object is an instance of a given class or a class derived from it:

```
if( $obj isa Package::Name ) { ... }
```

For more detail see "Class Instance Operator" in perlop.

Unicode 13.0 is supported

See <<https://www.unicode.org/versions/Unicode13.0.0/>> for details.

Chained comparisons capability

Some comparison operators, as their associativity, chain with some operators of the same precedence (but never with operators of different precedence).

```
if ( $x < $y <= $z ) {...}
```

behaves exactly like:

```
if ( $x < $y && $y <= $z ) {...}
```

(assuming that "\$y" is as simple a scalar as it looks.)

You can read more about this in perlop under "Operator Precedence and Associativity" in

perlop.

New Unicode properties "Identifier_Status" and "Identifier_Type" supported

Unicode has revised its regular expression requirements:

<<https://www.unicode.org/reports/tr18/tr18-21.html>>. As part of that they are wanting more properties to be exposed, ones that aren't part of the strict UCD (Unicode character database). These two are used for examining inputs for security purposes. Details on their usage is at <<https://www.unicode.org/reports/tr39/>>.

It is now possible to write "qr\p{Name=...}/", or "qr!\p{na=/(SMILING|GRINNING) FACE}/!"

The Unicode Name property is now accessible in regular expression patterns, as an alternative to "\N{...}". A comparison of the two methods is given in "Comparison of \N{...} and \p{name=...}" in perlunicode.

The second example above shows that wildcard subpatterns are also usable in this property.

See "Wildcards in Property Values" in perlunicode.

Improvement of "POSIX::mblen()", "mbtowc", and "wctomb"

The "POSIX::mblen()", "mbtowc", and "wctomb" functions now work on shift state locales and are thread-safe on C99 and above compilers when executed on a platform that has locale thread-safety; the length parameters are now optional.

These functions are always executed under the current C language locale. (See perllocale.) Most locales are stateless, but a few, notably the very rarely encountered ISO 2022, maintain a state between calls to these functions. Previously the state was cleared on every call, but now the state is not reset unless the appropriate parameter is "undef".

On threaded perls, the C99 functions mbrlen(3), mbrtowc(3), and wcrctomb(3), when available, are substituted for the plain functions. This makes these functions thread-safe when executing on a locale thread-safe platform.

The string length parameters in "mblen" and "mbtowc" are now optional; useful only if you wish to restrict the length parsed in the source string to less than the actual length.

Alpha assertions are no longer experimental

See "(*pla:pattern)" in perlre, "(*plb:pattern)" in perlre, "(*nla:pattern)" in perlre, and "(*nlb:pattern)" in perlre. Use of these no longer generates a warning; existing code that disables the warning category "experimental::alpha_assertions" will continue to work without any changes needed. Enabling the category has no effect.

Script runs are no longer experimental

See "Script Runs" in perlre. Use of these no longer generates a warning; existing code that disables the warning category "experimental::script_run" will continue to work without any changes needed. Enabling the category has no effect.

Feature checks are now faster

Previously feature checks in the parser required a hash lookup when features were set outside of a feature bundle, this has been optimized to a bit mask check. [GH #17229 <<https://github.com/Perl/perl5/issues/17229>>]

Perl is now developed on GitHub

Perl is now developed on GitHub. You can find us at <<https://github.com/Perl/perl5>>.

Non-security bugs should now be reported via GitHub. Security issues should continue to be reported as documented in perlsec.

Compiled patterns can now be dumped before optimization

This is primarily useful for tracking down bugs in the regular expression compiler. This dump happens on "-DDEBUGGING" perls, if you specify "-Drv" on the command line; or on any perl if the pattern is compiled within the scope of "use?re?qw(Debug?DUMP_PRE_OPTIMIZE)" or "use?re?qw(Debug?COMPILE?EXTRA)". (All but the second case display other information as well.)

Security

[CVE-2020-10543] Buffer overflow caused by a crafted regular expression

A signed "size_t" integer overflow in the storage space calculations for nested regular expression quantifiers could cause a heap buffer overflow in Perl's regular expression compiler that overwrites memory allocated after the regular expression storage space with attacker supplied data.

The target system needs a sufficient amount of memory to allocate partial expansions of the nested quantifiers prior to the overflow occurring. This requirement is unlikely to be met on 64-bit systems.

Discovered by: ManhND of The Tarantula Team, VinCSS (a member of Vingroup).

[CVE-2020-10878] Integer overflow via malformed bytecode produced by a crafted regular expression

Integer overflows in the calculation of offsets between instructions for the regular expression engine could cause corruption of the intermediate language state of a compiled regular expression. An attacker could abuse this behaviour to insert instructions into the compiled form of a Perl regular expression.

Discovered by: Hugo van der Sanden and Slaven Rezic.

[CVE-2020-12723] Buffer overflow caused by a crafted regular expression

Recursive calls to "S_study_chunk()" by Perl's regular expression compiler to optimize the intermediate language representation of a regular expression could cause corruption of the intermediate language state of a compiled regular expression.

Discovered by: Sergey Aleynikov.

Additional Note

An application written in Perl would only be vulnerable to any of the above flaws if it evaluates regular expressions supplied by the attacker. Evaluating regular expressions in this fashion is known to be dangerous since the regular expression engine does not protect against denial of service attacks in this usage scenario.

Incompatible Changes

Certain pattern matching features are now prohibited in compiling Unicode property value wildcard subpatterns

These few features are either inappropriate or interfere with the algorithm used to accomplish this task. The complete list is in "Wildcards in Property Values" in perlunicode.

Unused functions "POSIX::mbstowcs" and "POSIX::wcstombs" are removed

These functions could never have worked due to a defective interface specification. There is clearly no demand for them, given that no one has ever complained in the many years the functions were claimed to be available, hence so-called "support" for them is now dropped.

A bug fix for "(? [...])" may have caused some patterns to no longer compile

See "Selected Bug Fixes". The heuristics previously used may have let some constructs compile (perhaps not with the programmer's intended effect) that should have been errors. None are known, but it is possible that some erroneous constructs no longer compile.

"\p{user-defined}" properties now always override official Unicode ones

Previously, if and only if a user-defined property was declared prior to the compilation of the regular expression pattern that contains it, its definition was used instead of any official Unicode property with the same name. Now, it always overrides the official property. This change could break existing code that relied (likely unwittingly) on the previous behavior. Without this fix, if Unicode released a new version with a new property that happens to have the same name as the one you had long been using, your program would break when you upgraded to a perl that used that new Unicode version. See "User-Defined

Character Properties" in perlunicode. [GH #17205

<<https://github.com/Perl/perl5/issues/17205>>]

Modifiable variables are no longer permitted in constants

Code like:

```
my $var;  
  
$sub = sub () { $var };
```

where \$var is referenced elsewhere in some sort of modifiable context now produces an exception when the sub is defined.

This error can be avoided by adding a return to the sub definition:

```
$sub = sub () { return $var };
```

This has been deprecated since Perl 5.22. [GH #17020]

<<https://github.com/Perl/perl5/issues/17020>>

Use of "vec" on strings with code points above 0xFF is forbidden

Such strings are represented internally in UTF-8, and "vec" is a bit-oriented operation that will likely give unexpected results on those strings. This was deprecated in perl 5.28.0.

Use of code points over 0xFF in string bitwise operators

Some uses of these were already illegal after a previous deprecation cycle. The remaining uses are now prohibited, having been deprecated in perl 5.28.0. See perldeprecation.

"Sys::Hostname::hostname()" does not accept arguments

This usage was deprecated in perl 5.28.0 and is now fatal.

Plain "0" string now treated as a number for range operator

Previously a range "0" .. "-1" would produce a range of numeric strings from "0" through "99"; this now produces an empty list, just as "0 .. -1" does. This also means that "0" .. "9" now produces a list of integers, where previously it would produce a list of strings. This was due to a special case that treated strings starting with "0" as strings so ranges like "00" .. "03" produced "00", "01", "02", "03", but didn't specially handle the string "0". [GH #16770] <<https://github.com/Perl/perl5/issues/16770>>

"\K" now disallowed in look-ahead and look-behind assertions

This was disallowed because it causes unexpected behaviour, and no-one could define what the desired behaviour should be. [GH #14638] <<https://github.com/Perl/perl5/issues/14638>>

Performance Enhancements

? "my_strnlen" has been sped up for systems that don't have their own "strnlen"

implementation.

- ? "grok_bin_oct_hex" (and so, "grok_bin", "grok_oct", and "grok_hex") have been sped up.
- ? "grok_number_flags" has been sped up.
- ? "sort" is now noticeably faster in cases such as "sort {\$a <=> \$b}" or "sort {\$b <=> \$a}". [GH #17608 <<https://github.com/Perl/perl5/pull/17608>>]

Modules and Pragmata

Updated Modules and Pragmata

- ? Archive::Tar has been upgraded from version 2.32 to 2.36.
- ? autodie has been upgraded from version 2.29 to 2.32.
- ? B has been upgraded from version 1.76 to 1.80.
- ? B::Deparse has been upgraded from version 1.49 to 1.54.
- ? Benchmark has been upgraded from version 1.22 to 1.23.
- ? charnames has been upgraded from version 1.45 to 1.48.
- ? Class::Struct has been upgraded from version 0.65 to 0.66.
- ? Compress::Raw::Bzip2 has been upgraded from version 2.084 to 2.093.
- ? Compress::Raw::Zlib has been upgraded from version 2.084 to 2.093.
- ? CPAN has been upgraded from version 2.22 to 2.27.
- ? DB_File has been upgraded from version 1.843 to 1.853.
- ? Devel::PPPort has been upgraded from version 3.52 to 3.57.

The test files generated on Win32 are now identical to when they are generated on POSIX-like systems.

- ? diagnostics has been upgraded from version 1.36 to 1.37.
- ? Digest::MD5 has been upgraded from version 2.55 to 2.55_01.
- ? Dumpvalue has been upgraded from version 1.18 to 1.21.

Previously, when dumping elements of an array and encountering an undefined value, the string printed would have been "empty array". This has been changed to what was apparently originally intended: "empty slot".

- ? DynaLoader has been upgraded from version 1.45 to 1.47.
- ? Encode has been upgraded from version 3.01 to 3.06.
- ? encoding has been upgraded from version 2.22 to 3.00.
- ? English has been upgraded from version 1.10 to 1.11.
- ? Exporter has been upgraded from version 5.73 to 5.74.
- ? ExtUtils::CBuilder has been upgraded from version 0.280231 to 0.280234.

? ExtUtils::MakeMaker has been upgraded from version 7.34 to 7.44.

? feature has been upgraded from version 1.54 to 1.58.

A new "indirect" feature has been added, which is enabled by default but allows turning off indirect object syntax.

? File::Find has been upgraded from version 1.36 to 1.37.

On Win32, the tests no longer require either a file in the drive root directory, or a writable root directory.

? File::Glob has been upgraded from version 1.32 to 1.33.

? File::stat has been upgraded from version 1.08 to 1.09.

? Filter::Simple has been upgraded from version 0.95 to 0.96.

? Getopt::Long has been upgraded from version 2.5 to 2.51.

? Hash::Util has been upgraded from version 0.22 to 0.23.

The Synopsis has been updated as the example code stopped working with newer perls.

[GH #17399 <<https://github.com/Perl/perl5/issues/17399>>]

? I18N::Langinfo has been upgraded from version 0.18 to 0.19.

? I18N::LangTags has been upgraded from version 0.43 to 0.44.

Document the "IGNORE_WIN32_LOCALE" environment variable.

? IO has been upgraded from version 1.40 to 1.43.

IO::Socket no longer caches a zero protocol value, since this indicates that the implementation will select a protocol. This means that on platforms that don't implement "SO_PROTOCOL" for a given socket type the protocol method may return "undef".

The supplied TO is now always honoured on calls to the "send()" method. [GH #16891]

<<https://github.com/Perl/perl5/issues/16891>>

? IO-Compress has been upgraded from version 2.084 to 2.093.

? IPC::Cmd has been upgraded from version 1.02 to 1.04.

? IPC::Open3 has been upgraded from version 1.20 to 1.21.

? JSON::PP has been upgraded from version 4.02 to 4.04.

? Math::BigInt has been upgraded from version 1.999816 to 1.999818.

? Math::BigInt::FastCalc has been upgraded from version 0.5008 to 0.5009.

? Module::CoreList has been upgraded from version 5.20190522 to 5.20200620.

? Module::Load::Conditional has been upgraded from version 0.68 to 0.70.

? Module::Metadata has been upgraded from version 1.000036 to 1.000037.

- ? mro has been upgraded from version 1.22 to 1.23.
- ? Net::Ping has been upgraded from version 2.71 to 2.72.
- ? Opcode has been upgraded from version 1.43 to 1.47.
- ? open has been upgraded from version 1.11 to 1.12.
- ? overload has been upgraded from version 1.30 to 1.31.
- ? parent has been upgraded from version 0.237 to 0.238.
- ? perfaq has been upgraded from version 5.20190126 to 5.20200523.
- ? PerlIO has been upgraded from version 1.10 to 1.11.
- ? PerlIO::encoding has been upgraded from version 0.27 to 0.28.
- ? PerlIO::via has been upgraded from version 0.17 to 0.18.
- ? Pod::Html has been upgraded from version 1.24 to 1.25.
- ? Pod::Simple has been upgraded from version 3.35 to 3.40.
- ? podlators has been upgraded from version 4.11 to 4.14.
- ? POSIX has been upgraded from version 1.88 to 1.94.
- ? re has been upgraded from version 0.37 to 0.40.
- ? Safe has been upgraded from version 2.40 to 2.41.
- ? Scalar::Util has been upgraded from version 1.50 to 1.55.
- ? SelfLoader has been upgraded from version 1.25 to 1.26.
- ? Socket has been upgraded from version 2.027 to 2.029.
- ? Storable has been upgraded from version 3.15 to 3.21.

Use of "note()" from Test::More is now optional in tests. This works around a circular dependency with Test::More when installing on very old perls from CPAN.

Vstring magic strings over 2GB are now disallowed.

Regular expressions objects weren't properly counted for object id purposes on retrieve. This would corrupt the resulting structure, or cause a runtime error in some cases. [GH #17037] <<https://github.com/Perl/perl5/issues/17037>>

- ? Sys::Hostname has been upgraded from version 1.22 to 1.23.
- ? Sys::Syslog has been upgraded from version 0.35 to 0.36.
- ? Term::ANSIColor has been upgraded from version 4.06 to 5.01.
- ? Test::Simple has been upgraded from version 1.302162 to 1.302175.
- ? Thread has been upgraded from version 3.04 to 3.05.
- ? Thread::Queue has been upgraded from version 3.13 to 3.14.
- ? threads has been upgraded from version 2.22 to 2.25.

- ? threads::shared has been upgraded from version 1.60 to 1.61.
- ? Tie::File has been upgraded from version 1.02 to 1.06.
- ? Tie::Hash::NamedCapture has been upgraded from version 0.10 to 0.13.
- ? Tie::Scalar has been upgraded from version 1.04 to 1.05.
- ? Tie::StdHandle has been upgraded from version 4.5 to 4.6.
- ? Time::HiRes has been upgraded from version 1.9760 to 1.9764.

Removed obsolete code such as support for pre-5.6 perl and classic MacOS. [GH #17096]

<<https://github.com/Perl/perl5/issues/17096>>

- ? Time::Piece has been upgraded from version 1.33 to 1.3401.
- ? Unicode::Normalize has been upgraded from version 1.26 to 1.27.
- ? Unicode::UCD has been upgraded from version 0.72 to 0.75.
- ? VMS::Stdio has been upgraded from version 2.44 to 2.45.
- ? warnings has been upgraded from version 1.44 to 1.47.
- ? Win32 has been upgraded from version 0.52 to 0.53.
- ? Win32API::File has been upgraded from version 0.1203 to 0.1203_01.
- ? XS::APItest has been upgraded from version 1.00 to 1.09.

Removed Modules and Pragmata

- ? Pod::Parser has been removed from the core distribution. It still is available for download from CPAN. This resolves [#13194 <<https://github.com/Perl/perl5/issues/13194>>].

Documentation

Changes to Existing Documentation

We have attempted to update the documentation to reflect the changes listed in this document. If you find any we have missed, open an issue at <<https://github.com/Perl/perl5/issues>>.

Additionally, the following selected changes have been made:

perldebguts

- ? Simplify a few regnode definitions
Update "BOUND" and "NBOUND" definitions.

- ? Add ANYOFHs regnode

This node is like "ANYOFHb", but is used when more than one leading byte is the same in all the matched code points.

"ANYOFHb" is used to avoid having to convert from UTF-8 to code point for something

that won't match. It checks that the first byte in the UTF-8 encoded target is the desired one, thus ruling out most of the possible code points.

perlapi

- ? "sv_2pvbyte" updated to mention it will croak if the SV cannot be downgraded.
- ? "sv_setpvn" updated to mention that the UTF-8 flag will not be changed by this function, and a terminating NUL byte is guaranteed.
- ? Documentation for "PL_phase" has been added.
- ? The documentation for "grok_bin", "grok_oct", and "grok_hex" has been updated and clarified.

perldiag

- ? Add documentation for experimental 'isa' operator
(S experimental::isa) This warning is emitted if you use the ("isa") operator. This operator is currently experimental and its behaviour may change in future releases of Perl.

perlfunc

"caller"

Like "`__FILE__`" and "`__LINE__`", the filename and line number returned here may be altered by the mechanism described at "Plain Old Comments (Not!)" in `perlsyn`.

"`__FILE__`"

It can be altered by the mechanism described at "Plain Old Comments (Not!)" in `perlsyn`.

"`__LINE__`"

It can be altered by the mechanism described at "Plain Old Comments (Not!)" in `perlsyn`.

"return"

Now mentions that you cannot return from "do BLOCK".

"open"

The "open()" section had been renovated significantly.

perlguts

- ? No longer suggesting using perl's "malloc". Modern system "malloc" is assumed to be much better than perl's implementation now.
- ? Documentation about `embed.fnc` flags has been removed. `embed.fnc` now has sufficient comments within it. Anyone changing that file will see those comments first, so

entries here are now redundant.

? Updated documentation for "UTF8f"

? Added missing "=for apidoc" lines

perlhacktips

? The differences between Perl strings and C strings are now detailed.

perlintro

? The documentation for the repetition operator "x" have been clarified. [GH #17335
<<https://github.com/Perl/perl5/issues/17335>>]

perlipc

? The documentation surrounding "open" and handle usage has been modernized to prefer 3-arg open and lexical variables instead of barewords.

? Various updates and fixes including making all examples strict-safe and replacing "-w" with "use warnings".

perlop

? 'isa' operator is experimental

This is an experimental feature and is available when enabled by "use feature 'isa'".

It emits a warning in the "experimental::isa" category.

perlpod

? Details of the various stacks within the perl interpreter are now explained here.

? Advice has been added regarding the usage of "Z<>".

perlport

? Update "timegm" example to use the correct year format 1970 instead of 70. [GH #16431
<<https://github.com/Perl/perl5/issues/16431>>]

perlrefref

? Fix some typos.

perlvar

? Now recommends stringifying \$] and comparing it numerically.

perlapi, perlintern

? Documentation has been added for several functions that were lacking it before.

perlx

? Suggest using "libffi" for simple library bindings via CPAN modules like FFI::Platypus or FFI::Raw.

POSIX

? "setlocale" warning about threaded builds updated to note it does not apply on Perl 5.28.X and later.

? "Posix::SigSet->new(...)" updated to state it throws an error if any of the supplied signals cannot be added to the set.

Additionally, the following selected changes have been made:

Updating of links

? Links to the now defunct <<https://search.cpan.org>> site now point at the equivalent <<https://metacpan.org>> URL. [GH #17393 <<https://github.com/Perl/perl5/issues/17393>>]

? The man page for ExtUtils::XSymSet is now only installed on VMS, which is the only platform the module is installed on. [GH #17424 <<https://github.com/Perl/perl5/issues/17424>>]

? URLs have been changed to "https://" and stale links have been updated.

Where applicable, the URLs in the documentation have been moved from using the "http://" protocol to "https://". This also affects the location of the bug tracker at <<https://rt.perl.org>>.

? Some links to OS/2 libraries, Address Sanitizer and other system tools had gone stale. These have been updated with working links.

? Some links to old email addresses on perl5-porters had gone stale. These have been updated with working links.

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.

New Diagnostics

New Errors

? Expecting interpolated extended charclass in regex; marked by <-- HERE in m/%s/
This is a replacement for several error messages listed under "Changes to Existing Diagnostics".

? "No digits found for %s literal"

(F) No hexadecimal digits were found following "0x" or no binary digits were found following "0b".

New Warnings

? Code point 0x%X is not Unicode, and not portable

This is actually not a new message, but it is now output when the warnings category

"portable" is enabled.

When raised during regular expression pattern compilation, the warning has extra text added at the end marking where precisely in the pattern it occurred.

? Non-hex character '%c' terminates \x early. Resolved as "%s"

This replaces a warning that was much less specific, and which gave false information.

This new warning parallels the similar already-existing one raised for "\o{}".

Changes to Existing Diagnostics

? Character following "\c" must be printable ASCII

...now has extra text added at the end, when raised during regular expression pattern compilation, marking where precisely in the pattern it occurred.

? Use "%s" instead of "%s"

...now has extra text added at the end, when raised during regular expression pattern compilation, marking where precisely in the pattern it occurred.

? Sequence "\c{" invalid

...now has extra text added at the end, when raised during regular expression pattern compilation, marking where precisely in the pattern it occurred.

? "\c%c" is more clearly written simply as "%s"

...now has extra text added at the end, when raised during regular expression pattern compilation, marking where precisely in the pattern it occurred.

? Non-octal character '%c' terminates \o early. Resolved as "%s"

...now includes the phrase "terminates \o early", and has extra text added at the end, when raised during regular expression pattern compilation, marking where precisely in the pattern it occurred. In some instances the text of the resolution has been clarified.

? '%s' resolved to '\o{%s}%d'

As of Perl 5.32, this message is no longer generated. Instead, "Non-octal character '%c' terminates \o early. Resolved as "%s"" in perldiag is used instead.

? Use of code point 0x%s is not allowed; the permissible max is 0x%X

Some instances of this message previously output the hex digits "A", "B", "C", "D", "E", and "F" in lower case. Now they are all consistently upper case.

? The following three diagnostics have been removed, and replaced by "Expecting interpolated extended charclass in regex; marked by <-- HERE in m/%s/" : "Expecting close paren for nested extended charclass in regex; marked by <-- HERE in m/%s/",

"Expecting close paren for wrapper for nested extended charclass in regex; marked by <-- HERE in m/%s/", and "Expecting '(?flags:(?['...' in regex; marked by <--?HERE in m/%s/".

? The "Code point 0x%X is not Unicode, and not portable" warning removed the line "Code points above 0xFFFF_FFFF require larger than a 32 bit word." as code points that large are no longer legal on 32-bit platforms.

? Can't use global %s in %s

This error message has been slightly reformatted from the original "Can't use global %s in "%s"", and in particular misleading error messages like "Can't use global \$_ in "my"" are now rendered as "Can't use global \$_ in subroutine signature".

? Constants from lexical variables potentially modified elsewhere are no longer permitted

This error message replaces the former "Constants from lexical variables potentially modified elsewhere are deprecated. This will not be allowed in Perl 5.32" to reflect the fact that this previously deprecated usage has now been transformed into an exception. The message's classification has also been updated from D (deprecated) to F (fatal).

See also "Incompatible Changes".

? "\N{} here is restricted to one character" is now emitted in the same circumstances where previously "\N{} in inverted character class or as a range end-point is restricted to one character" was.

This is due to new circumstances having been added in Perl 5.30 that weren't covered by the earlier wording.

Utility Changes

perlbug

? The bug tracker homepage URL now points to GitHub.

streamzip

? This is a new utility, included as part of an IO::Compress::Base upgrade.

streamzip creates a zip file from stdin. The program will read data from stdin, compress it into a zip container and, by default, write a streamed zip file to stdout.

Configuration and Compilation

Configure

? For clang++, add "#include <stdlib.h>" to Configure's probes for "futimes", "strtoll",

"strtol", "strtoull", "strtouq", otherwise the probes would fail to compile.

- ? Use a compile and run test for "lchown" to satisfy clang++ which should more reliably detect it.
- ? For C++ compilers, add "#include <stdio.h>" to Configure's probes for "getpgrp" and "setpgrp" as they use printf and C++ compilers may fail compilation instead of just warning.
- ? Check if the compiler can handle inline attribute.
- ? Check for character data alignment.
- ? Configure now correctly handles gcc-10. Previously it was interpreting it as gcc-1 and turned on "-fpcc-struct-return".
- ? Perl now no longer probes for "d_u32align", defaulting to "define" on all platforms. This check was error-prone when it was done, which was on 32-bit platforms only. [GH #16680] <<https://github.com/Perl/perl5/issues/16680>>
- ? Documentation and hints for building perl on Z/OS (native EBCDIC) have been updated. This is still a work in progress.
- ? A new probe for "malloc_usable_size" has been added.
- ? Improvements in Configure to detection in C++ and clang++. Work ongoing by Andy Dougherty. [GH #17033] <<https://github.com/Perl/perl5/issues/17033>>
- ? autodoc.pl
This tool that regenerates perlintern and perlapi has been overhauled significantly, restoring consistency in flags used in embed.fnc and Devel::PPPort and allowing removal of many redundant "=for apidoc" entries in code.
- ? The "ECHO" macro is now defined. This is used in a "dtrace" rule that was originally changed for FreeBSD, and the FreeBSD make apparently predefines it. The Solaris make does not predefine "ECHO" which broke this rule on Solaris. [GH #17057] <<https://github.com/Perl/perl5/issues/17057>>
- ? Bison versions 3.1 through 3.4 are now supported.

Testing

Tests were added and changed to reflect the other additions and changes in this release.

Furthermore, these significant changes were made:

- ? t/run/switches.t no longer uses (and re-uses) the tmpinplace/ directory under t/. This may prevent spurious failures. [GH #17424] <<https://github.com/Perl/perl5/issues/17424>>

- ? Various bugs in "POSIX::mbtowc" were fixed. Potential races with other threads are now avoided, and previously the returned wide character could well be garbage.
- ? Various bugs in "POSIX::wctomb" were fixed. Potential races with other threads are now avoided, and previously it would segfault if the string parameter was shared or hadn't been pre-allocated with a string of sufficient length to hold the result.
- ? Certain test output of scalars containing control characters and Unicode has been fixed on EBCDIC.
- ? t/charset_tools.pl: Avoid some work on ASCII platforms.
- ? t/re/regexp.t: Speed up many regex tests on ASCII platform
- ? t/re/pat.t: Skip tests that don't work on EBCDIC.

Platform Support

Discontinued Platforms

Windows CE

Support for building perl on Windows CE has now been removed.

Platform-Specific Notes

Linux

"cc" will be used to populate "plibpth" if "cc" is "clang". [GH #17043]

<<https://github.com/Perl/perl5/issues/17043>>

NetBSD 8.0

Fix compilation of Perl on NetBSD 8.0 with g++. [GH #17381]

<<https://github.com/Perl/perl5/issues/17381>>

Windows

? The configuration for "cflags" and "optimize" are now separate, as with POSIX platforms. [GH #17156 <<https://github.com/Perl/perl5/issues/17156>>]

? Support for building perl with Visual C++ 6.0 has now been removed.

? The locale tests could crash on Win32 due to a Windows bug, and separately due to the CRT throwing an exception if the locale name wasn't validly encoded in the current code page.

For the second we now decode the locale name ourselves, and always decode it as UTF-8. [GH #16922] <<https://github.com/Perl/perl5/issues/16922>>

? t/op/magic.t could fail if environment variables starting with "FOO" already existed.

? MYMALLOC (PERL_MALLOC) build has been fixed.

Solaris

- ? "Configure" will now find recent versions of the Oracle Developer Studio compiler, which are found under `"/opt/developerstudio*"`.
- ? "Configure" now uses the detected types for `"gethostby*"` functions, allowing Perl to once again compile on certain configurations of Solaris.

VMS

- ? With the release of the patch kit C99 V2.0, VSI has provided support for a number of previously-missing C99 features. On systems with that patch kit installed, Perl's configuration process will now detect the presence of the header `"stdint.h"` and the following functions: `"fpclassify"`, `"isblank"`, `"isless"`, `"llrint"`, `"llrintl"`, `"llround"`, `"llroundl"`, `"nearbyint"`, `"round"`, `"scalbn"`, and `"scalbnl"`.
- ? `"-Duse64bitint"` is now the default on VMS.

z/OS

Perl 5.32 has been tested on z/OS 2.4, with the following caveats:

- ? Only static builds (the default) build reliably
- ? When using locales, z/OS does not handle the `"LC_MESSAGES"` category properly, so when compiling perl, you should add the following to your Configure options
`./Configure <other options> -Accflags=-DNO_LOCALE_MESSAGES`
- ? z/OS does not support locales with threads, so when compiling a threaded perl, you should add the following to your Configure options
`./Configure <other Configure options> -Accflags=-DNO_LOCALE`
- ? Some CPAN modules that are shipped with perl fail at least one of their self-tests. These are: `Archive::Tar`, `Config::Perl::V`, `CPAN::Meta`, `CPAN::Meta::YAML`, `Digest::MD5`, `Digest::SHA`, `Encode`, `ExtUtils::MakeMaker`, `ExtUtils::Manifest`, `HTTP::Tiny`, `IO::Compress`, `IPC::Cmd`, `JSON::PP`, `libnet`, `MIME::Base64`, `Module::Metadata`, `PerlIO::via-QuotedPrint`, `Pod::Checker`, `podlators`, `Pod::Simple`, `Socket`, and `Test::Harness`.

The causes of the failures range from the self-test itself is flawed, and the module actually works fine, up to the module doesn't work at all on EBCDIC platforms.

Internal Changes

- ? `"savepv"`'s `len` parameter is now a `"Size_t"` instead of an `"I32"` since we can handle longer strings than 31 bits.

- ? The lexer ("Perl_yylex()" in `toke.c`) was previously a single 4100-line function, relying heavily on "goto" and a lot of widely-scoped local variables to do its work. It has now been pulled apart into a few dozen smaller static functions; the largest remaining chunk ("`yy_l_word_or_keyword()`") is a little over 900 lines, and consists of a single "switch" statement, all of whose "case" groups are independent. This should be much easier to understand and maintain.
- ? The OS-level signal handlers and type (`Sighandler_t`) used by the perl core were declared as having three parameters, but the OS was always told to call them with one argument. This has been fixed by declaring them to have one parameter. See the merge commit "`v5.31.5-346-g116e19abbf`" for full details.
- ? The code that handles "`tr//`" has been extensively revised, fixing various bugs, especially when the source and/or replacement strings contain characters whose code points are above 255. Some of the bugs were undocumented, one being that under some circumstances (but not all) with "`/s`", the squeezing was done based on the source, rather than the replacement. A documented bug that got fixed was [GH #14777] <https://github.com/Perl/perl5/issues/14777>.
- ? A new macro for XS writers dealing with UTF-8-encoded Unicode strings has been created "`UTF8_CHK_SKIP`" in `perlapi` that is safer in the face of malformed UTF-8 input than "`UTF8_SKIP`" in `perlapi` (but not as safe as "`UTF8_SAFE_SKIP`" in `perlapi`). It won't read past a NUL character. It has been backported in `Devel::PPPort 3.55` and later.
- ? Added the "`PL_curstackinfo->si_cxsubix`" field. This records the stack index of the most recently pushed sub/format/eval context. It is set and restored automatically by "`cx_pushsub()`", "`cx_popsub()`" etc., but would need to be manually managed if you do any unusual manipulation of the context stack.
- ? Various macros dealing with character type classification and changing case where the input is encoded in UTF-8 now require an extra parameter to prevent potential reads beyond the end of the buffer. Use of these has generated a deprecation warning since Perl 5.26. Details are in "In XS code, use of various macros dealing with UTF-8." in `perldeprecation`
- ? A new parser function `parse_subsignature()` allows a keyword plugin to parse a subroutine signature while "`use feature 'signatures'`" is in effect. This allows custom keywords to implement semantics similar to regular "sub" declarations that include signatures. [GH #16261] <https://github.com/Perl/perl5/issues/16261>

- ? Since on some platforms we need to hold a mutex when temporarily switching locales, new macros ("STORE_LC_NUMERIC_SET_TO_NEEDED_IN", "WITH_LC_NUMERIC_SET_TO_NEEDED" and "WITH_LC_NUMERIC_SET_TO_NEEDED_IN") have been added to make it easier to do this safely and efficiently as part of [GH #17034] <<https://github.com/Perl/perl5/issues/17034>>.
- ? The memory bookkeeping overhead for allocating an OP structure has been reduced by 8 bytes per OP on 64-bit systems.
- ? eval_pv() no longer stringifies the exception when "[GH #17035]" <<https://github.com/Perl/perl5/issues/17035>>]
- ? The PERL_DESTRUCT_LEVEL environment variable was formerly only honoured on perl binaries built with DEBUGGING support. It is now checked on all perl builds. Its normal use is to force perl to individually free every block of memory which it has allocated before exiting, which is useful when using automated leak detection tools such as valgrind.
- ? The API eval_sv() now accepts a "G_RETHROW" flag. If this flag is set and an exception is thrown while compiling or executing the supplied code, it will be rethrown, and eval_sv() will not return. [GH #17036] <<https://github.com/Perl/perl5/issues/17036>>
- ? As part of the fix for [GH #1537] <<https://github.com/Perl/perl5/issues/1537>> perl_parse() now returns non-zero if exit(0) is called in a "BEGIN", "UNITCHECK" or "CHECK" block.
- ? Most functions which recursively walked an op tree during compilation have been made non-recursive. This avoids SEGVs from stack overflow when the op tree is deeply nested, such as "\$n == 1 ? "one" : \$n == 2 ? "two" :" (especially in code which is auto-generated).
This is particularly noticeable where the code is compiled within a separate thread, as threads tend to have small stacks by default.

Selected Bug Fixes

- ? Previously "require" in perlfunc would only treat the special built-in SV &PL_sv_undef as a value in %INC as if a previous "require" has failed, treating other undefined SVs as if the previous "require" has succeeded. This could cause unexpected success from "require" e.g., on "local %INC = %INC;". This has been fixed. [GH #17428] <<https://github.com/Perl/perl5/issues/17428>>]
- ? "{...}" eval groups in regular expressions no longer unintentionally trigger "EVAL

- without pos change exceeded limit in regex" [GH #17490
<<https://github.com/Perl/perl5/issues/17490>>].
- ? "(?[\...])" extended bracketed character classes do not wrongly raise an error on some cases where a previously-compiled such class is interpolated into another. The heuristics previously used have been replaced by a reliable method, and hence the diagnostics generated have changed. See "Diagnostics".
- ? The debug display (say by specifying "-Dr" or "use?re" (with appropriate options) of compiled Unicode property wildcard subpatterns no longer has extraneous output.
- ? Fix an assertion failure in the regular expression engine. [GH #17372
<<https://github.com/Perl/perl5/issues/17372>>]
- ? Fix coredump in pp_hot.c after "B::UNOP_AUX::aux_list()". [GH #17301
<<https://github.com/Perl/perl5/issues/17301>>]
- ? Loading IO is now threadsafe. [GH #14816
<<https://github.com/Perl/perl5/issues/14816>>]
- ? "\p{user-defined}" overrides official Unicode [GH #17025
<<https://github.com/Perl/perl5/issues/17025>>]
Prior to this patch, the override was only sometimes in effect.
- ? Properly handle filled "/il" regnodes and multi-char folds
- ? Compilation error during make minitest [GH #17293
<<https://github.com/Perl/perl5/issues/17293>>]
- ? Move the implementation of "%-", "%+" into core.
- ? Read beyond buffer in "grok_inf_nan" [GH #17370
<<https://github.com/Perl/perl5/issues/17370>>]
- ? Workaround glibc bug with "LC_MESSAGES" [GH #17081
<<https://github.com/Perl/perl5/issues/17081>>]
- ? "printf()" or "sprintf()" with the %n format could cause a panic on debugging builds, or report an incorrectly cached length value when producing "SVfUTF8" flagged strings.
[GH #17221 <<https://github.com/Perl/perl5/issues/17221>>]
- ? The tokenizer has been extensively refactored. [GH #17241
<<https://github.com/Perl/perl5/issues/17241>>] [GH #17189
<<https://github.com/Perl/perl5/issues/17189>>]
- ? "use strict "subs"" is now enforced for bareword constants optimized into a "multiconcat" operator. [GH #17254 <<https://github.com/Perl/perl5/issues/17254>>]

- ? A memory leak in regular expression patterns has been fixed. [GH #17218
<<https://github.com/Perl/perl5/issues/17218>>]
- ? Perl no longer treats strings starting with "0x" or "0b" as hex or binary numbers respectively when converting a string to a number. This reverts a change in behaviour inadvertently introduced in perl 5.30.0 intended to improve precision when converting a string to a floating point number. [GH #17062]
<<https://github.com/Perl/perl5/issues/17062>>
- ? Matching a non-"SVf_UTF8" string against a regular expression containing unicode literals could leak a SV on each match attempt. [GH #17140]
<<https://github.com/Perl/perl5/issues/17140>>
- ? Overloads for octal and binary floating point literals were always passed a string with a "0x" prefix instead of the appropriate 0 or "[GH #14791]]<https://github.com/Perl/perl5/issues/14791>]"
- ? "\$@ = 100; die;" now correctly propagates the 100 as an exception instead of ignoring it. [GH #17098] <<https://github.com/Perl/perl5/issues/17098>>
- ? "[GH #17108]]<https://github.com/Perl/perl5/issues/17108>]"
- ? Exceptions thrown while \$@ is read-only could result in infinite recursion as perl tried to update \$@, which throws another exception, resulting in a stack overflow. Perl now replaces \$@ with a copy if it's not a simple writable SV. [GH #17083]
<<https://github.com/Perl/perl5/issues/17083>>
- ? Setting \$) now properly sets supplementary group ids if you have the necessary privileges. [GH #17031] <<https://github.com/Perl/perl5/issues/17031>>
- ? close() on a pipe now preemptively clears the PerlIO object from the IO SV. This prevents a second attempt to close the already closed PerlIO object if a signal handler calls die() or exit() while close() is waiting for the child process to complete. [GH #13929] <<https://github.com/Perl/perl5/issues/13929>>
- ? "sprintf("%.a", -10000, \$x)" would cause a buffer overflow due to mishandling of the negative precision value. [GH #16942] <<https://github.com/Perl/perl5/issues/16942>>
- ? scalar() on a reference could cause an erroneous assertion failure during compilation. [GH #16969] <<https://github.com/Perl/perl5/issues/16969>>
- ? "%{^CAPTURE_ALL}" is now an alias to "%-" as documented, rather than incorrectly an alias for "[GH #16105]]<https://github.com/Perl/perl5/issues/16105>]"
- ? "%{^CAPTURE}" didn't work if "@{^CAPTURE}" was mentioned first. Similarly for

- "%{^CAPTURE_ALL}" and "@{^CAPTURE_ALL}", though "[GH #17045]|<https://github.com/Perl/perl5/issues/17045>"]
- ? Extraordinarily large (over 2GB) floating point format widths could cause an integer overflow in the underlying call to `sprintf()`, resulting in an assertion. Formatted floating point widths are now limited to the range of `int`, the return value of `sprintf()`. [GH #16881 <<https://github.com/Perl/perl5/issues/16881>>]
 - ? Parsing the following constructs within a sub-parse (such as with "`{code here}`" or "`s/.../code here/e`") has changed to match how they're parsed normally:
 - ? "`print $fh ...`" no longer produces a syntax error.
 - ? Code like "`s/.../ ${time} /e`" now properly produces an "Ambiguous use of `{time}` resolved to `$time` at ..." warning when warnings are enabled.
 - ? "`@x {"a"}`" (with the space) in a sub-parse now properly produces a "better written as" warning when warnings are enabled.
 - ? Attributes can now be used in a sub-parse. [GH #16847 <<https://github.com/Perl/perl5/issues/16847>>]
 - ? Incomplete hex and binary literals like "`0x`" and "`0b`" are now treated as if the "`x`" or "`b`" is part of the next token. [GH #17010 <<https://github.com/Perl/perl5/issues/17010>>]
 - ? A spurious ")" in a subparse, such as in "`s/.../code here/e`" or "`...${code here}`", no longer confuses the parser.

Previously a subparse was bracketed with generated "(" and ")" tokens, so a spurious ")" would close the construct without doing the normal subparse clean up, confusing the parser and possibly causing an assertion failure.

Such constructs are now surrounded by artificial tokens that can't be included in the source. [GH #15814] <<https://github.com/Perl/perl5/issues/15814>>
 - ? Reference assignment of a sub, such as "`&foo = \&bar;`", silently did nothing in the past. [GH #16987]|<https://github.com/Perl/perl5/issues/16987>"]
 - ? `sv_gets()` now recovers better if the target SV is modified by a signal handler. [GH #16960] <<https://github.com/Perl/perl5/issues/16960>>
 - ? "`readline @foo`" now evaluates `@foo` in scalar context. Previously it would be evaluated in list context, and since `readline()` pops only one argument from the stack, the stack could underflow, or be left with unexpected values on the stack. [GH #16929] <<https://github.com/Perl/perl5/issues/16929>>
 - ? Parsing incomplete hex or binary literals was changed in 5.31.1 to treat such a

literal as just the 0, leaving the following "x" or "b" to be parsed as part of the next token. This could lead to some silent changes in behaviour, so now incomplete hex or binary literals produce a fatal error. [GH #17010]

<<https://github.com/Perl/perl5/issues/17010>>

? eval_pv()'s croak_on_error flag will now throw even if the exception is a false overloaded value. [GH #17036] <<https://github.com/Perl/perl5/issues/17036>>

? "INIT" blocks and the program itself are no longer run if exit(0) is called within a "BEGIN", "UNITCHECK" or "CHECK" block. [GH #1537]

<<https://github.com/Perl/perl5/issues/1537>>

? "open my \$fh, ">>+", undef" now opens the temporary file in append mode: writes will seek to the end of file before writing. [GH #17058]

<<https://github.com/Perl/perl5/issues/17058>>

? Fixed a SEGV when searching for the source of an uninitialized value warning on an op whose subtree includes an OP_MULTIDEREF. [GH #17088]

<<https://github.com/Perl/perl5/issues/17088>>

Obituary

Jeff Goff (JGOFF or DrForr), an integral part of the Perl and Raku communities and a dear friend to all of us, has passed away on March 13th, 2020. DrForr was a prominent member of the communities, attending and speaking at countless events, contributing to numerous projects, and assisting and helping in any way he could.

His passing leaves a hole in our hearts and in our communities and he will be sorely missed.

Acknowledgements

Perl 5.32.0 represents approximately 13 months of development since Perl 5.30.0 and contains approximately 220,000 lines of changes across 1,800 files from 89 authors.

Excluding auto-generated files, documentation and release tools, there were approximately 140,000 lines of changes to 880 .pm, .t, .c and .h files.

Perl continues to flourish into its fourth decade thanks to a vibrant community of users and developers. The following people are known to have contributed the improvements that became Perl 5.32.0:

Aaron Crane, Alberto Simões, Alexandr Savca, Andreas K?nig, Andrew Fresh, Andy Dougherty, Ask Bj?rn Hansen, Atsushi Sugawara, Bernhard M. Wiedemann, brian d foy, Bryan Stenson, Chad Granum, Chase Whitener, Chris 'BinGOs' Williams, Craig A. Berry, Dagfinn Ilmari

Manns?ker, Dan Book, Daniel Dragan, Dan Kogai, Dave Cross, Dave Rolsky, David Cantrell, David Mitchell, Dominic Hargreaves, E. Choroba, Felipe Gasper, Florian Weimer, Graham Knop, H?kon H?gland, Hauke D, H.Merijn Brand, Hugo van der Sanden, Ichinose Shogo, James E Keenan, Jason McIntosh, Jerome Duval, Johan Vromans, John Lightsey, John Paul Adrian Glaubitz, Kang-min Liu, Karen Etheridge, Karl Williamson, Leon Timmermans, Manuel Mausz, Marc Green, Matthew Horsfall, Matt Turner, Max Maischein, Michael Haardt, Nicholas Clark, Nicolas R., Niko Tyni, Pali, Paul Evans, Paul Johnson, Paul Marquess, Peter Eisentraut, Peter John Acklam, Peter Oliver, Petr P?sa?, Renee Baecker, Ricardo Signes, Richard Leach, Russ Allbery, Samuel Smith, Santtu Ojanper?, Sawyer X, Sergey Aleynikov, Sergiy Borodych, Shirakata Kentaro, Shlomi Fish, Sisyphus, Slaven Rezic, Smylers, Stefan Seifert, Steve Hay, Steve Peters, Svyatoslav, Thibault Duponchelle, Todd Rinaldo, Tomasz Konojacki, Tom Hukins, Tony Cook, Unicode Consortium, VanL, Vickenty Fesunov, Vitali Peil, Yves Orton, Zefram.

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 perl bug database at

<<https://github.com/Perl/perl5/issues>>. There may also be information at

<<http://www.perl.org/>>, the Perl Home Page.

If you believe you have an unreported bug, please open an issue at

<<https://github.com/Perl/perl5/issues>>. Be sure to trim your bug down to a tiny but

sufficient test case.

If the bug you are reporting has security implications which make it inappropriate to send

to a public issue tracker, then see "SECURITY VULNERABILITY CONTACT INFORMATION" in

perlsec for details of how to report the issue.

Give Thanks

If you wish to thank the Perl 5 Porters for the work we had done in Perl 5, you can do so

by running the "perlthanks" program:

perlthanks

This will send an email to the Perl 5 Porters list with your show of thanks.

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.

perl v5.34.0

2023-11-23

PERL5320DELTA(1)