NAME

perl5300delta - what is new for perl v5.30.0

DESCRIPTION

This document describes differences between the 5.28.0 release and the 5.30.0 release.

If you are upgrading from an earlier release such as 5.26.0, first read perl5280delta, which describes differences between 5.26.0 and 5.28.0.

Notice

sv_utf8_(downgrade|decode) are no longer marked as experimental. [GH #16822] <https://github.com/Perl/perl5/issues/16822>.

Core Enhancements

Limited variable length lookbehind in regular expression pattern matching is now experimentally supported

Using a lookbehind assertion (like "(?<=foo?)" or "(?<!ba{1,9}r)" previously would generate an error and refuse to compile. Now it compiles (if the maximum lookbehind is at most 255 characters), but raises a warning in the new "experimental::vlb" warnings category. This is to caution you that the precise behavior is subject to change based on feedback from use in the field.

See "(?<=pattern)" in perlre and "(?<!pattern)" in perlre.

The upper limit "n" specifiable in a regular expression quantifier of the form "{m,n}" has been doubled to 65534

The meaning of an unbounded upper quantifier "{m,}" remains unchanged. It matches $2^{31} - 1$ times on most platforms, and more on ones where a C language short variable is more than 4 bytes long.

Unicode 12.1 is supported

Because of a change in Unicode release cycles, Perl jumps from Unicode 10.0 in Perl 5.28 to Unicode 12.1 in Perl 5.30.

For details on the Unicode changes, see <https://www.unicode.org/versions/Unicode11.0.0/> for 11.0; <https://www.unicode.org/versions/Unicode12.0.0/> for 12.0; and <https://www.unicode.org/versions/Unicode12.1.0/> for 12.1. (Unicode 12.1 differs from 12.0 only in the addition of a single character, that for the new Japanese era name.)

The Word_Break property, as in past Perl releases, remains tailored to behave more in line with expectations of Perl users. This means that sequential runs of horizontal white space characters are not broken apart, but kept as a single run. Unicode 11 changed from past versions to be more in line with

Perl, but it left several white space characters as causing breaks: TAB, NO BREAK SPACE, and FIGURE SPACE (U+2007). We have decided to continue to use the previous Perl tailoring with regards to these.

Wildcards in Unicode property value specifications are now partially supported
You can now do something like this in a regular expression pattern

  qr! \p{nv= /(?x) \A [0-5] \z / }!

which matches all Unicode code points whose numeric value is between 0 and 5 inclusive. So, it could match the Thai or Bengali digits whose numeric values are 0, 1, 2, 3, 4, or 5.

This marks another step in implementing the regular expression features the Unicode Consortium suggests.

Most properties are supported, with the remainder planned for 5.32. Details are in "Wildcards in Property Values" in perlunicode.

qr'\N{name}' is now supported
Previously it was an error to evaluate a named character "\N{...}" within a single quoted regular expression pattern (whose evaluation is deferred from the normal place). This restriction is now removed.

Turkic UTF-8 locales are now seamlessly supported
Turkic languages have different casing rules than other languages for the characters "i" and "I". The uppercase of "i" is LATIN CAPITAL LETTER I WITH DOT ABOVE (U+0130); and the lowercase of "I" is LATIN SMALL LETTER DOTLESS I (U+0131). Unicode furnishes alternate casing rules for use with Turkic languages. Previously, Perl ignored these, but now, it uses them when it detects that it is operating under a Turkic UTF-8 locale.

It is now possible to compile perl to always use thread-safe locale operations.
Previously, these calls were only used when the perl was compiled to be multi-threaded. To always enable them, add

  -Accflags='-DUSE_THREAD_SAFE_LOCALE'

to your *Configure* flags.

Eliminate opASSIGN macro usage from core
This macro is still defined but no longer used in core

"-Drv" now means something on "-DDEBUGGING" builds

Now, adding the verbose flag ("-Dv") to the "-Dr" flag turns on all possible regular expression debugging.

Incompatible Changes

Assigning non-zero to $[is fatal

Setting $[to a non-zero value has been deprecated since Perl 5.12 and now throws a fatal error. See "Assigning non-zero to $[is fatal" in perldeprecation.

Delimiters must now be graphemes

See "Use of unassigned code point or non-standalone grapheme for a delimiter." in perldeprecation

Some formerly deprecated uses of an unescaped left brace "{" in regular expression patterns are now illegal

But to avoid breaking code unnecessarily, most instances that issued a deprecation warning, remain legal and now have a non-deprecation warning raised. See "Unescaped left braces in regular expressions" in perldeprecation.

Previously deprecated sysread()/syswrite() on :utf8 handles is now fatal

Calling **sysread()**, **syswrite()**, **send()** or **recv()** on a ":utf8" handle, whether applied explicitly or implicitly, is now fatal. This was deprecated in perl 5.24.

There were two problems with calling these functions on ":utf8" handles:

⊕   All four functions only paid attention to the ":utf8" flag. Other layers were completely ignored, so a handle with ":encoding(UTF-16LE)" layer would be treated as UTF-8. Other layers, such as compression are completely ignored with or without the ":utf8" flag.

⊕   **sysread()** and **recv()** would read from the handle, skipping any validation by the layers, and do no validation of their own. This could lead to invalidly encoded perl scalars.

[GH #14839] <https://github.com/Perl/perl5/issues/14839>.

my() in false conditional prohibited

Declarations such as "my $x if 0" are no longer permitted.

[GH #16702] <https://github.com/Perl/perl5/issues/16702>.

Fatalize $* and $#

These special variables, long deprecated, now throw exceptions when used.

[GH #16718] <https://github.com/Perl/perl5/issues/16718>.

Fatalize unqualified use of dump()

The "dump()" function, long discouraged, may no longer be used unless it is fully qualified, *i.e.*, "CORE::dump()".

[GH #16719] <https://github.com/Perl/perl5/issues/16719>.

Remove File::Glob::glob()

The "File::Glob::glob()" function, long deprecated, has been removed and now throws an exception which advises use of "File::Glob::bsd_glob()" instead.

[GH #16721] <https://github.com/Perl/perl5/issues/16721>.

"pack()" no longer can return malformed UTF-8

It croaks if it would otherwise return a UTF-8 string that contains malformed UTF-8. This protects against potential security threats. This is considered a bug fix as well. [GH #16035] <https://github.com/Perl/perl5/issues/16035>.

Any set of digits in the Common script are legal in a script run of another script

There are several sets of digits in the Common script. "[0-9]" is the most familiar. But there are also "[\x{FF10}-\x{FF19}]" (FULLWIDTH DIGIT ZERO - FULLWIDTH DIGIT NINE), and several sets for use in mathematical notation, such as the MATHEMATICAL DOUBLE-STRUCK DIGITs. Any of these sets should be able to appear in script runs of, say, Greek. But the design of 5.30 overlooked all but the ASCII digits "[0-9]", so the design was flawed. This has been fixed, so is both a bug fix and an incompatibility. [GH #16704] <https://github.com/Perl/perl5/issues/16704>.

All digits in a run still have to come from the same set of ten digits.

JSON::PP enables allow_nonref by default

As JSON::XS 4.0 changed its policy and enabled allow_nonref by default, JSON::PP also enabled allow_nonref by default.

Deprecations

In XS code, use of various macros dealing with UTF-8.

This deprecation was scheduled to become fatal in 5.30, but has been delayed to 5.32 due to problems that showed up with some CPAN modules. For details of what's affected, see perldeprecation.

Performance Enhancements

⊕   Translating from UTF-8 into the code point it represents now is done via a deterministic finite

automaton, speeding it up. As a typical example, "ord("\x7fff")" now requires 12% fewer instructions than before. The performance of checking that a sequence of bytes is valid UTF-8 is similarly improved, again by using a DFA.

⊕ Eliminate recursion from **finalize_op()**. [GH #11866] <https://github.com/Perl/perl5/issues/11866>.

⊕ A handful of small optimizations related to character folding and character classes in regular expressions.

⊕ Optimization of "IV" to "UV" conversions. [GH #16761] <https://github.com/Perl/perl5/issues/16761>.

⊕ Speed up of the integer stringification algorithm by processing two digits at a time instead of one. [GH #16769] <https://github.com/Perl/perl5/issues/16769>.

⊕ Improvements based on LGTM analysis and recommendation. (<https://lgtm.com/projects/g/Perl/perl5/alerts/?mode=tree>). [GH #16765] <https://github.com/Perl/perl5/issues/16765>. [GH #16773] <https://github.com/Perl/perl5/issues/16773>.

⊕ Code optimizations in *regcomp.c*, *regcomp.h*, *regexec.c*.

⊕ Regular expression pattern matching of things like "qr/[^*a*]/" is significantly sped up, where *a* is any ASCII character. Other classes can get this speed up, but which ones is complicated and depends on the underlying bit patterns of those characters, so differs between ASCII and EBCDIC platforms, but all case pairs, like "qr/[Gg]/" are included, as is "[^01]".

Modules and Pragmata

 Updated Modules and Pragmata
⊕ Archive::Tar has been upgraded from version 2.30 to 2.32.

⊕ B has been upgraded from version 1.74 to 1.76.

⊕ B::Concise has been upgraded from version 1.003 to 1.004.

⊕ B::Deparse has been upgraded from version 1.48 to 1.49.

⊕ bignum has been upgraded from version 0.49 to 0.51.

⊕   bytes has been upgraded from version 1.06 to 1.07.

⊕   Carp has been upgraded from version 1.38 to 1.50

⊕   Compress::Raw::Bzip2 has been upgraded from version 2.074 to 2.084.

⊕   Compress::Raw::Zlib has been upgraded from version 2.076 to 2.084.

⊕   Config::Extensions has been upgraded from version 0.02 to 0.03.

⊕   Config::Perl::V. has been upgraded from version 0.29 to 0.32. This was due to a new
     configuration variable that has influence on binary compatibility:
     "USE_THREAD_SAFE_LOCALE".

⊕   CPAN has been upgraded from version 2.20 to 2.22.

⊕   Data::Dumper has been upgraded from version 2.170 to 2.174

     Data::Dumper now avoids leaking when "croak"ing.

⊕   DB_File has been upgraded from version 1.840 to 1.843.

⊕   deprecate has been upgraded from version 0.03 to 0.04.

⊕   Devel::Peek has been upgraded from version 1.27 to 1.28.

⊕   Devel::PPPort has been upgraded from version 3.40 to 3.52.

⊕   Digest::SHA has been upgraded from version 6.01 to 6.02.

⊕   Encode has been upgraded from version 2.97 to 3.01.

⊕   Errno has been upgraded from version 1.29 to 1.30.

⊕   experimental has been upgraded from version 0.019 to 0.020.

⊕   ExtUtils::CBuilder has been upgraded from version 0.280230 to 0.280231.

⊕   ExtUtils::Manifest has been upgraded from version 1.70 to 1.72.

⊕   ExtUtils::Miniperl has been upgraded from version 1.08 to 1.09.

⊕   ExtUtils::ParseXS has been upgraded from version 3.39 to 3.40. "OUTLIST" parameters are no longer incorrectly included in the automatically generated function prototype. [GH #16746] <https://github.com/Perl/perl5/issues/16746>.

⊕   feature has been upgraded from version 1.52 to 1.54.

⊕   File::Copy has been upgraded from version 2.33 to 2.34.

⊕   File::Find has been upgraded from version 1.34 to 1.36.

    $File::Find::dont_use_nlink now defaults to 1 on all platforms. [GH #16759] <https://github.com/Perl/perl5/issues/16759>.

    Variables $Is_Win32 and $Is_VMS are being initialized.

⊕   File::Glob has been upgraded from version 1.31 to 1.32.

⊕   File::Path has been upgraded from version 2.15 to 2.16.

⊕   File::Spec has been upgraded from version 3.74 to 3.78.

    Silence Cwd warning on Android builds if "targetsh" is not defined.

⊕   File::Temp has been upgraded from version 0.2304 to 0.2309.

⊕   Filter::Util::Call has been upgraded from version 1.58 to 1.59.

⊕   GDBM_File has been upgraded from version 1.17 to 1.18.

⊕   HTTP::Tiny has been upgraded from version 0.070 to 0.076.

⊕   I18N::Langinfo has been upgraded from version 0.17 to 0.18.

⊕   IO has been upgraded from version 1.39 to 1.40.

⊕   IO-Compress has been upgraded from version 2.074 to 2.084.

    Adds support for "IO::Uncompress::Zstd" and "IO::Uncompress::UnLzip".

The "BinModeIn" and "BinModeOut" options are now no-ops. ALL files will be read/written in binmode.

⊕   IPC::Cmd has been upgraded from version 1.00 to 1.02.

⊕   JSON::PP has been upgraded from version 2.97001 to 4.02.

    JSON::PP as JSON::XS 4.0 enables "allow_nonref" by default.

⊕   lib has been upgraded from version 0.64 to 0.65.

⊕   Locale::Codes has been upgraded from version 3.56 to 3.57.

⊕   Math::BigInt has been upgraded from version 1.999811 to 1.999816.

    "bnok()" now supports the full Kronenburg extension. [cpan #95628]
    <https://rt.cpan.org/Ticket/Display.html?id=95628>.

⊕   Math::BigInt::FastCalc has been upgraded from version 0.5006 to 0.5008.

⊕   Math::BigRat has been upgraded from version 0.2613 to 0.2614.

⊕   Module::CoreList has been upgraded from version 5.20180622 to 5.20190520.

    Changes to B::Op_private and Config

⊕   Module::Load has been upgraded from version 0.32 to 0.34.

⊕   Module::Metadata has been upgraded from version 1.000033 to 1.000036.

    Properly clean up temporary directories after testing.

⊕   NDBM_File has been upgraded from version 1.14 to 1.15.

⊕   Net::Ping has been upgraded from version 2.62 to 2.71.

⊕   ODBM_File has been upgraded from version 1.15 to 1.16.

⊕   PathTools has been upgraded from version 3.74 to 3.78.

⊕   parent has been upgraded from version 0.236 to 0.237.

⊕   perl5db.pl has been upgraded from version 1.54 to 1.55.

    Debugging threaded code no longer deadlocks in "DB::sub" nor "DB::lsub".

⊕   perlfaq has been upgraded from version 5.021011 to 5.20190126.

⊕   PerlIO::encoding has been upgraded from version 0.26 to 0.27.

    Warnings enabled by setting the "WARN_ON_ERR" flag in $PerlIO::encoding::fallback are now
    only produced if warnings are enabled with "use warnings "utf8";" or setting $^W.

⊕   PerlIO::scalar has been upgraded from version 0.29 to 0.30.

⊕   podlators has been upgraded from version 4.10 to 4.11.

⊕   POSIX has been upgraded from version 1.84 to 1.88.

⊕   re has been upgraded from version 0.36 to 0.37.

⊕   SDBM_File has been upgraded from version 1.14 to 1.15.

⊕   sigtrap has been upgraded from version 1.08 to 1.09.

⊕   Storable has been upgraded from version 3.08 to 3.15.

    Storable no longer probes for recursion limits at build time. [GH #16780]
    <https://github.com/Perl/perl5/issues/16780> and others.

    Metasploit exploit code was included to test for CVE-2015-1992 detection, this caused anti-virus
    detections on at least one AV suite. The exploit code has been removed and replaced with a
    simple functional test. [GH #16778] <https://github.com/Perl/perl5/issues/16778>

⊕   Test::Simple has been upgraded from version 1.302133 to 1.302162.

⊕   Thread::Queue has been upgraded from version 3.12 to 3.13.

⊕   threads::shared has been upgraded from version 1.58 to 1.60.

Added support for extra tracing of locking, this requires a "-DDEBUGGING" and extra compilation flags.

⊕   Time::HiRes has been upgraded from version 1.9759 to 1.9760.

⊕   Time::Local has been upgraded from version 1.25 to 1.28.

⊕   Time::Piece has been upgraded from version 1.3204 to 1.33.

⊕   Unicode::Collate has been upgraded from version 1.25 to 1.27.

⊕   Unicode::UCD has been upgraded from version 0.70 to 0.72.

⊕   User::grent has been upgraded from version 1.02 to 1.03.

⊕   utf8 has been upgraded from version 1.21 to 1.22.

⊕   vars has been upgraded from version 1.04 to 1.05.

   "vars.pm" no longer disables non-vars strict when checking if strict vars is enabled. [GH #15851] <https://github.com/Perl/perl5/issues/15851>.

⊕   version has been upgraded from version 0.9923 to 0.9924.

⊕   warnings has been upgraded from version 1.42 to 1.44.

⊕   XS::APItest has been upgraded from version 0.98 to 1.00.

⊕   XS::Typemap has been upgraded from version 0.16 to 0.17.

Removed Modules and Pragmata

The following modules will be removed from the core distribution in a future release, and will at that time need to be installed from CPAN. Distributions on CPAN which require these modules will need to list them as prerequisites.

The core versions of these modules will now issue "deprecated"-category warnings to alert you to this fact. To silence these deprecation warnings, install the modules in question from CPAN.

Note that these are (with rare exceptions) fine modules that you are encouraged to continue to use. Their disinclusion from core primarily hinges on their necessity to bootstrapping a fully functional,

CPAN-capable Perl installation, not usually on concerns over their design.

⊕   B::Debug is no longer distributed with the core distribution. It continues to be available on CPAN as "B::Debug <https://metacpan.org/pod/B::Debug>".

⊕   Locale::Codes has been removed at the request of its author. It continues to be available on CPAN as "Locale::Codes <https://metacpan.org/pod/Locale::Codes>" [GH #16660] <https://github.com/Perl/perl5/issues/16660>.

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, send email to perlbug@perl.org <mailto:perlbug@perl.org>.

   perlapi

⊕   "AvFILL()" was wrongly listed as deprecated. This has been corrected. [GH #16586] <https://github.com/Perl/perl5/issues/16586>

   perlop

⊕   We no longer have null (empty line) here doc terminators, so perlop should not refer to them.

⊕   The behaviour of "tr" when the delimiter is an apostrophe has been clarified. In particular, hyphens aren't special, and "\x{}" isn't interpolated. [GH #15853] <https://github.com/Perl/perl5/issues/15853>

   perlreapi, perlvar

⊕   Improve docs for lastparen, lastcloseparen.

   perlfunc

⊕   The entry for "-X" in perlfunc has been clarified to indicate that symbolic links are followed for most tests.

⊕   Clarification of behaviour of "reset EXPR".

⊕   Try to clarify that "ref(qr/xx/)" returns "Regexp" rather than "REGEXP" and why. [GH #16801] <https://github.com/Perl/perl5/issues/16801>.

perlreref

⊕   Clarification of the syntax of /(?(cond)yes)/.

perllocale

⊕   There are actually two slightly different types of UTF-8 locales: one for Turkic languages and one
    for everything else. Starting in Perl v5.30, Perl seamlessly handles both types.

perlrecharclass

⊕   Added a note for the ::xdigit:: character class.

perlvar

⊕   More specific documentation of paragraph mode. [GH #16787]
    <https://github.com/Perl/perl5/issues/16787>.

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

⊕   As noted under "Incompatible Changes" above, the deprecation warning "Unescaped left brace in
    regex is deprecated here (and will be fatal in Perl 5.30), passed through in regex; marked by
    <-- HERE in m/%s/" has been changed to the non-deprecation warning "Unescaped left brace in
    regex is passed through in regex; marked by <-- HERE in m/%s/".

⊕   Specifying "\o{}" without anything between the braces now yields the fatal error message "Empty
    \o{}". Previously it was "Number with no digits". This means the same wording is used for this
    kind of error as with similar constructs such as "\p{}".

⊕   Within the scope of the experimental feature "use re 'strict'", specifying "\x{}" without anything
    between the braces now yields the fatal error message "Empty \x{}". Previously it was "Number
    with no digits". This means the same wording is used for this kind of error as with similar
    constructs such as "\p{}". It is legal, though not wise to have an empty "\x" outside of "re
    'strict'"; it silently generates a NUL character.

⊕   Type of arg %d to %s must be %s (not %s)

Attempts to push, pop, etc on a hash or glob now produce this message rather than complaining that they no longer work on scalars. [GH #15774] <https://github.com/Perl/perl5/issues/15774>.

⊕   Prototype not terminated

The file and line number is now reported for this error. [GH #16697] <https://github.com/Perl/perl5/issues/16697>

⊕   Under "-Dr" (or "use re 'Debug'") the compiled regex engine program is displayed. It used to use two different spellings for *infinity*, "INFINITY", and "INFTY". It now uses the latter exclusively, as that spelling has been around the longest.

Utility Changes
xsubpp
⊕   The generated prototype (with "PROTOTYPES: ENABLE") would include "OUTLIST" parameters, but these aren't arguments to the perl function. This has been rectified. [GH #16746] <https://github.com/Perl/perl5/issues/16746>.

Configuration and Compilation
⊕   Normally the thread-safe locale functions are used only on threaded builds. It is now possible to force their use on unthreaded builds on systems that have them available, by including the "-Accflags='-DUSE_THREAD_SAFE_LOCALE'" option to *Configure*.

⊕   Improve detection of memrchr, strlcat, and strlcpy

⊕   Improve Configure detection of **memmem()**. [GH #16807] <https://github.com/Perl/perl5/issues/16807>.

⊕   Multiple improvements and fixes for -DPERL_GLOBAL_STRUCT build option.

⊕   Fix -DPERL_GLOBAL_STRUCT_PRIVATE build option.

Testing
⊕   *t/lib/croak/op* [GH #15774] <https://github.com/Perl/perl5/issues/15774>.

separate error for "push", etc. on hash/glob.

⊕   *t/op/svleak.t* [GH #16749] <https://github.com/Perl/perl5/issues/16749>.

Add test for "goto &sub" in overload leaking.

⊕ Split *t/re/fold_grind.t* into multiple test files.

⊕ Fix intermittent tests which failed due to race conditions which surface during parallel testing.
[GH #16795] <https://github.com/Perl/perl5/issues/16795>.

⊕ Thoroughly test paragraph mode, using a new test file, *t/io/paragraph_mode.t*. [GH #16787]
<https://github.com/Perl/perl5/issues/16787>.

⊕ Some tests in *t/io/eintr.t* caused the process to hang on pre-16 Darwin. These tests are skipped for
those version of Darwin.

Platform Support
Platform-Specific Notes
HP-UX 11.11
> An obscure problem in "pack()" when compiling with HP C-ANSI-C has been fixed by disabling
> optimizations in *pp_pack.c*.

Mac OS X
> Perl's build and testing process on Mac OS X for "-Duseshrplib" builds is now compatible with
> Mac OS X System Integrity Protection (SIP).
>
> SIP prevents binaries in */bin* (and a few other places) being passed the
> "DYLD_LIBRARY_PATH" environment variable. For our purposes this prevents
> "DYLD_LIBRARY_PATH" from being passed to the shell, which prevents that variable being
> passed to the testing or build process, so running "perl" couldn't find *libperl.dylib*.
>
> To work around that, the initial build of the *perl* executable expects to find *libperl.dylib* in the
> build directory, and the library path is then adjusted during installation to point to the installed
> library.
>
> [GH #15057] <https://github.com/Perl/perl5/issues/15057>.

Minix3
> Some support for Minix3 has been re-added.

Cygwin
> Cygwin doesn't make "cuserid" visible.

Win32 Mingw
> C99 math functions are now available.

Windows

⊕   The "USE_CPLUSPLUS" build option which has long been available in *win32/Makefile* (for
      nmake) and *win32/makefile.mk* (for **dmake**) is now also available in *win32/GNUmakefile*
      (for **gmake**).

⊕   The **nmake** makefile no longer defaults to Visual
      C

      6.0 (a very old version which is unlikely to be widely used today). As a result, it is now a
      requirement to specify the "CCTYPE" since there is no obvious choice of which modern
      version to default to instead. Failure to specify "CCTYPE" will result in an error being
      output and the build will stop.

      (The **dmake** and **gmake** makefiles will automatically detect which compiler is being used, so
      do not require "CCTYPE" to be set. This feature has not yet been added to the **nmake**
      makefile.)

⊕   "sleep()" with warnings enabled for a "USE_IMP_SYS" build no longer warns about the
      sleep timeout being too large. [GH #16631] <https://github.com/Perl/perl5/issues/16631>.

⊕   Support for compiling perl on Windows using Microsoft Visual Studio 2019 (containing
      Visual
      C

      14.2) has been added.

⊕   **socket()** now sets $! if the protocol, address family and socket type combination is not found.
      [GH #16849] <https://github.com/Perl/perl5/issues/16849>.

⊕   The Windows Server 2003 SP1 Platform SDK build, with its early x64 compiler and tools,
      was accidentally broken in Perl 5.27.9. This has now been fixed.

Internal Changes

⊕   The sizing pass has been eliminated from the regular expression compiler. An extra pass may
      instead be needed in some cases to count the number of parenthetical capture groups.

⊕   A new function ""my_strtod"" in perlapi or its synonym, **Strtod**(), is now available with the same
      signature as the libc **strtod**(). It provides **strotod**() equivalent behavior on all platforms, using the
      best available precision, depending on platform capabilities and *Configure* options, while handling
      locale-related issues, such as if the radix character should be a dot or comma.

⊕   Added "newSVsv_nomg()" to copy a SV without processing get magic on the source. [GH #16461] <https://github.com/Perl/perl5/issues/16461>.

⊕   It is now forbidden to malloc more than "PTRDIFF_T_MAX" bytes. Much code (including C optimizers) assumes that all data structures will not be larger than this, so this catches such attempts before overflow happens.

⊕   Two new regnodes have been introduced "EXACT_ONLY8", and "EXACTFU_ONLY8". They're equivalent to "EXACT" and "EXACTFU", except that they contain a code point which requires UTF-8 to represent/match. Hence, if the target string isn't UTF-8, we know it can't possibly match, without needing to try.

⊕   "print_bytes_for_locale()" is now defined if "DEBUGGING", Prior, it didn't get defined unless "LC_COLLATE" was defined on the platform.

Selected Bug Fixes

⊕   Compilation under "-DPERL_MEM_LOG" and "-DNO_LOCALE" have been fixed.

⊕   Perl 5.28 introduced an "index()" optimization when comparing to -1 (or indirectly, e.g. >= 0). When this optimization was triggered inside a "when" clause it caused a warning ("Argument %s isn't numeric in smart match"). This has now been fixed. [GH #16626] <https://github.com/Perl/perl5/issues/16626>

⊕   The new in-place editing code no longer leaks directory handles. [GH #16602] <https://github.com/Perl/perl5/issues/16602>.

⊕   Warnings produced from constant folding operations on overloaded values no longer produce spurious "Use of uninitialized value" warnings. [GH #16349] <https://github.com/Perl/perl5/issues/16349>.

⊕   Fix for "mutator not seen in (lex = ...) .= ..." [GH #16655] <https://github.com/Perl/perl5/issues/16655>.

⊕   "pack "u", "invalid uuencoding"" now properly NUL terminates the zero-length SV produced. [GH #16343] <https://github.com/Perl/perl5/issues/16343>.

⊕   Improve the debugging output for **calloc()** calls with "-Dm". [GH #16653] <https://github.com/Perl/perl5/issues/16653>.

⊕   Regexp script runs were failing to permit ASCII digits in some cases. [GH #16704]

<https://github.com/Perl/perl5/issues/16704>.

⊕   On Unix-like systems supporting a platform-specific technique for determining $^X, Perl failed to
    fall back to the generic technique when the platform-specific one fails (for example, a Linux
    system with /proc not mounted). This was a regression in Perl 5.28.0. [GH #16715]
    <https://github.com/Perl/perl5/issues/16715>.

⊕   SDBM_File is now more robust with corrupt database files. The improvements do not make
    SDBM files suitable as an interchange format. [GH #16164]
    <https://github.com/Perl/perl5/issues/16164>.

⊕   "binmode($fh);" or "binmode($fh, ':raw');" now properly removes the ":utf8" flag from the
    default ":crlf" I/O layer on Win32. [GH #16730] <https://github.com/Perl/perl5/issues/16730>.

⊕   The experimental reference aliasing feature was misinterpreting array and hash slice assignment
    as being localised, e.g.

        \(@a[3,5,7]) = \(....);

    was being interpreted as:

        local \(@a[3,5,7]) = \(....);

    [GH #16701] <https://github.com/Perl/perl5/issues/16701>.

⊕   "sort SUBNAME" within an "eval EXPR" when "EXPR" was UTF-8 upgraded could panic if the
    "SUBNAME" was non-ASCII. [GH #16979] <https://github.com/Perl/perl5/issues/16979>.

⊕   Correctly handle **realloc()** modifying "errno" on success so that the modification isn't visible to
    the perl user, since **realloc()** is called implicitly by the interpreter. This modification is permitted
    by the C standard, but has only been observed on FreeBSD 13.0-CURRENT. [GH #16907]
    <https://github.com/Perl/perl5/issues/16907>.

⊕   Perl now exposes POSIX "getcwd" as "Internals::getcwd()" if available. This is intended for use
    by "Cwd.pm" during bootstrapping and may be removed or changed without notice. This fixes
    some bootstrapping issues while building perl in a directory where some ancestor directory isn't
    readable. [GH #16903] <https://github.com/Perl/perl5/issues/16903>.

⊕   "pack()" no longer can return malformed UTF-8. It croaks if it would otherwise return a UTF-8
    string that contains malformed UTF-8. This protects against potential security threats. [GH

#16035] <https://github.com/Perl/perl5/issues/16035>.

⊕   See "Any set of digits in the Common script are legal in a script run of another script".

⊕   Regular expression matching no longer leaves stale UTF-8 length magic when updating $^R. This
    could result in "length($^R)" returning an incorrect value.

⊕   Reduce recursion on ops [GH #11866] <https://github.com/Perl/perl5/issues/11866>.

    This can prevent stack overflow when processing extremely deep op trees.

⊕   Avoid leak in multiconcat with overloading. [GH #16823]
    <https://github.com/Perl/perl5/issues/16823>.

⊕   The handling of user-defined "\p{}" properties (see "User-Defined Character Properties" in
    perlunicode) has been rewritten to be in C (instead of Perl). This speeds things up, but in the
    process several inconsistencies and bug fixes are made.

    1.   A few error messages have minor wording changes. This is essentially because the new way
         is integrated into the regex error handling mechanism that marks the position in the input at
         which the error occurred. That was not possible previously. The messages now also contain
         additional back-trace-like information in case the error occurs deep in nested calls.

    2.   A user-defined property is implemented as a perl subroutine with certain highly constrained
         naming conventions. It was documented previously that the sub would be in the current
         package if the package was unspecified. This turned out not to be true in all cases, but now it
         is.

    3.   All recursive calls are treated as infinite recursion. Previously they would cause the
         interpreter to panic. Now, they cause the regex pattern to fail to compile.

    4.   Similarly, any other error likely would lead to a panic; now to just the pattern failing to
         compile.

    5.   The old mechanism did not detect illegal ranges in the definition of the property. Now, the
         range max must not be smaller than the range min. Otherwise, the pattern fails to compile.

    6.   The intention was to have each sub called only once during the lifetime of the program, so
         that a property's definition is immutable. This was relaxed so that it could be called once for
         all /i compilations, and potentially a second time for non-/i (the sub is passed a parameter

indicating which). However, in practice there were instances when this was broken, and multiple calls were possible. Those have been fixed. Now (besides the /i,non-/i cases) the only way a sub can be called multiple times is if some component of it has not been defined yet. For example, suppose we have sub **IsA()** whose definition is known at compile time, and it in turn calls **isB()** whose definition is not yet known. **isA()** will be called each time a pattern it appears in is compiled. If **isA()** also calls **isC()** and that definition is known, **isC()** will be called just once.

7.   There were some races and very long hangs should one thread be compiling the same property as another simultaneously. These have now been fixed.

⊕   Fixed a failure to match properly.

An EXACTFish regnode has a finite length it can hold for the string being matched. If that length is exceeded, a second node is used for the next segment of the string, for as many regnodes as are needed. Care has to be taken where to break the string, in order to deal multi-character folds in Unicode correctly. If we want to break a string at a place which could potentially be in the middle of a multi-character fold, we back off one (or more) characters, leaving a shorter EXACTFish regnode. This backing off mechanism contained an off-by-one error. [GH #16806] <https://github.com/Perl/perl5/issues/16806>.

⊕   A bare "eof" call with no previous file handle now returns true. [GH #16786] <https://github.com/Perl/perl5/issues/16786>

⊕   Failing to compile a format now aborts compilation. Like other errors in sub-parses this could leave the parser in a strange state, possibly crashing perl if compilation continued. [GH #16169] <https://github.com/Perl/perl5/issues/16169>

⊕   If an in-place edit is still in progress during global destruction and the process exit code (as stored in $?) is zero, perl will now treat the in-place edit as successful, replacing the input file with any output produced.

This allows code like:

```
perl -i -ne 'print "Foo"; last'
```

to replace the input file, while code like:

```
perl -i -ne 'print "Foo"; die'
```

will not. Partly resolves [GH #16748] <https://github.com/Perl/perl5/issues/16748>.

⊕   A regression in 5.28 caused the following code to fail

   close(STDIN); open(CHILD, "|wc -l")'

because the child's stdin would be closed on exec. This has now been fixed.

⊕   Fixed an issue where compiling a regexp containing both compile-time and run-time code blocks could lead to trying to compile something which is invalid syntax.

⊕   Fixed build failures with "-DNO_LOCALE_NUMERIC" and "-DNO_LOCALE_COLLATE".
    [GH #16771] <https://github.com/Perl/perl5/issues/16771>.

⊕   Prevent the tests in *ext/B/t/strict.t* from being skipped. [GH #16783]
    <https://github.com/Perl/perl5/issues/16783>.

⊕   "/di" nodes ending or beginning in *s* are now "EXACTF". We do not want two "EXACTFU" to be joined together during optimization, and to form a "ss", "sS", "Ss" or "SS" sequence; they are the only multi-character sequences which may match differently under "/ui" and "/di".

Acknowledgements

Perl 5.30.0 represents approximately 11 months of development since Perl 5.28.0 and contains approximately 620,000 lines of changes across 1,300 files from 58 authors.

Excluding auto-generated files, documentation and release tools, there were approximately 510,000 lines of changes to 750 .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 5.30.0:

Aaron Crane, Abigail, Alberto Simo~es, Alexandr Savca, Andreas Koenig, Andy Dougherty, Aristotle Pagaltzis, Brian Greenfield, Chad Granum, Chris 'BinGOs' Williams, Craig A. Berry, Dagfinn Ilmari Mannsaaker, Dan Book, Dan Dedrick, Daniel Dragan, Dan Kogai, David Cantrell, David Mitchell, Dominic Hargreaves, E. Choroba, Ed J, Eugen Konkov, Francois Perrad, Graham Knop, Hauke D, H.Merijn Brand, Hugo van der Sanden, Jakub Wilk, James Clarke, James E Keenan, Jerry D. Hedden, Jim Cromie, John SJ Anderson, Karen Etheridge, Karl Williamson, Leon Timmermans, Matthias Bethke, Nicholas Clark, Nicolas R., Niko Tyni, Pali, Petr PisaX, Phil Pearl (Lobbes), Richard Leach, Ryan Voots, Sawyer X, Shlomi Fish, Sisyphus, Slaven Rezic, Steve Hay, Sullivan Beck, Tina Mueller, Tomasz Konojacki, Tom Wyant, Tony Cook, Unicode Consortium, Yves Orton, Zak B. Elep.

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 most of the (very much appreciated) contributors who reported issues to the Perl bug tracker. Noteworthy in this release were the large number of bug fixes made possible by Sergey Aleynikov's high quality perlbug reports for issues he discovered by fuzzing with AFL.

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://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 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.