

NAME

perl5280delta - what is new for perl v5.28.0

DESCRIPTION

This document describes differences between the 5.26.0 release and the 5.28.0 release.

If you are upgrading from an earlier release such as 5.24.0, first read perl5260delta, which describes differences between 5.24.0 and 5.26.0.

Core Enhancements**Unicode 10.0 is supported**

A list of changes is at <<http://www.unicode.org/versions/Unicode10.0.0>>.

"delete" on key/value hash slices

"delete" can now be used on key/value hash slices, returning the keys along with the deleted values. [GH #15982] <<https://github.com/Perl/perl5/issues/15982>>

Experimentally, there are now alphabetic synonyms for some regular expression assertions

If you find it difficult to remember how to write certain of the pattern assertions, there are now alphabetic synonyms.

CURRENT	NEW SYNONYMS
-----	-----
(?=...)	(*pla:...) or (*positive_lookahead:...)
(?!...)	(*nla:...) or (*negative_lookahead:...)
(?<=...)	(*plb:...) or (*positive_lookbehind:...)
(?<!...)	(*nlb:...) or (*negative_lookbehind:...)
(?>...)	(*atomic:...)

These are considered experimental, so using any of these will raise (unless turned off) a warning in the "experimental::alpha_assertions" category.

Mixed Unicode scripts are now detectable

A mixture of scripts, such as Cyrillic and Latin, in a string is often the sign of a spoofing attack. A new regular expression construct now allows for easy detection of these. For example, you can say

```
qr/( *script_run: \d+ \b )/x
```

And the digits matched will all be from the same set of 10. You won't get a look-alike digit from a different script that has a different value than what it appears to be.

Or:

```
qr/(*sr: \b \w+ \b )/x
```

makes sure that all the characters come from the same script.

You can also combine script runs with "(?>...)" (or "*atomic:...").

Instead of writing:

```
(*sr:(?<...))
```

you can now run:

```
(*asr:...)
```

```
# or
```

```
(*atomic_script_run:...)
```

This is considered experimental, so using it will raise (unless turned off) a warning in the "experimental::script_run" category.

See "Script Runs" in perlre.

In-place editing with "perl -i" is now safer

Previously in-place editing ("perl -i") would delete or rename the input file as soon as you started working on a new file.

Without backups this would result in loss of data if there was an error, such as a full disk, when writing to the output file.

This has changed so that the input file isn't replaced until the output file has been completely written and successfully closed.

This works by creating a work file in the same directory, which is renamed over the input file once the output file is complete.

Incompatibilities:

- ⊕ Since this renaming needs to only happen once, if you create a thread or child process, that renaming will only happen in the original thread or process.

- ⊕ If you change directories while processing a file, and your operating system doesn't provide the "unlinkat()", "renameat()" and "fchmodat()" functions, the final rename step may fail.

[GH #15216] <<https://github.com/Perl/perl5/issues/15216>>

Initialisation of aggregate state variables

A persistent lexical array or hash variable can now be initialized, by an expression such as "state @a = qw(x y z)". Initialization of a list of persistent lexical variables is still not possible.

Full-size inode numbers

On platforms where inode numbers are of a type larger than perl's native integer numerical types, stat will preserve the full content of large inode numbers by returning them in the form of strings of decimal digits. Exact comparison of inode numbers can thus be achieved by comparing with "eq" rather than "==". Comparison with "==", and other numerical operations (which are usually meaningless on inode numbers), work as well as they did before, which is to say they fall back to floating point, and ultimately operate on a fairly useless rounded inode number if the real inode number is too big for the floating point format.

The "sprintf" %j format size modifier is now available with pre-C99 compilers

The actual size used depends on the platform, so remains unportable.

Close-on-exec flag set atomically

When opening a file descriptor, perl now generally opens it with its close-on-exec flag already set, on platforms that support doing so. This improves thread safety, because it means that an "exec" initiated by one thread can no longer cause a file descriptor in the process of being opened by another thread to be accidentally passed to the executed program.

Additionally, perl now sets the close-on-exec flag more reliably, whether it does so atomically or not. Most file descriptors were getting the flag set, but some were being missed.

String- and number-specific bitwise ops are no longer experimental

The new string-specific ("&. |. ^. ~.") and number-specific ("& | ^ ~") bitwise operators introduced in Perl 5.22 that are available within the scope of "use feature 'bitwise'" are no longer experimental. Because the number-specific ops are spelled the same way as the existing operators that choose their behaviour based on their operands, these operators must still be enabled via the "bitwise" feature, in either of these two ways:

```
use feature "bitwise";
```

```
use v5.28; # "bitwise" now included
```

They are also now enabled by the **-E** command-line switch.

The "bitwise" feature no longer emits a warning. Existing code that disables the "experimental::bitwise" warning category that the feature previously used will continue to work.

One caveat that module authors ought to be aware of is that the numeric operators now pass a fifth TRUE argument to overload methods. Any methods that check the number of operands may croak if they do not expect so many. XS authors in particular should be aware that this:

```
SV *
bitop_handler (lobj, robj, swap)
```

may need to be changed to this:

```
SV *
bitop_handler (lobj, robj, swap, ...)
```

Locales are now thread-safe on systems that support them

These systems include Windows starting with Visual Studio 2005, and in POSIX 2008 systems.

The implication is that you are now free to use locales and change them in a threaded environment. Your changes affect only your thread. See "Multi-threaded operation" in perllocale

New read-only predefined variable "\${^SAFE_LOCALES}"

This variable is 1 if the Perl interpreter is operating in an environment where it is safe to use and change locales (see perllocale.) This variable is true when the perl is unthreaded, or compiled in a platform that supports thread-safe locale operation (see previous item).

Security

[CVE-2017-12837] Heap buffer overflow in regular expression compiler

Compiling certain regular expression patterns with the case-insensitive modifier could cause a heap buffer overflow and crash perl. This has now been fixed. [GH #16021]
<<https://github.com/Perl/perl5/issues/16021>>

[CVE-2017-12883] Buffer over-read in regular expression parser

For certain types of syntax error in a regular expression pattern, the error message could either contain the contents of a random, possibly large, chunk of memory, or could crash perl. This has now been fixed. [GH #16025] <<https://github.com/Perl/perl5/issues/16025>>

[CVE-2017-12814] \$ENV{\$key} stack buffer overflow on Windows

A possible stack buffer overflow in the %ENV code on Windows has been fixed by removing the buffer completely since it was superfluous anyway. [GH #16051]
<<https://github.com/Perl/perl5/issues/16051>>

Default Hash Function Change

Perl 5.28.0 retires various older hash functions which are not viewed as sufficiently secure for use in Perl. We now support four general purpose hash functions, Siphash (2-4 and 1-3 variants), and Zaphod32, and StadtX hash. In addition we support SBOX32 (a form of tabular hashing) for hashing short strings, in conjunction with any of the other hash functions provided.

By default Perl is configured to support SBOX hashing of strings up to 24 characters, in conjunction with StadtX hashing on 64 bit builds, and Zaphod32 hashing for 32 bit builds.

You may control these settings with the following options to Configure:

```
-DPERL_HASH_FUNC_SIPHASH  
-DPERL_HASH_FUNC_SIPHASH13  
-DPERL_HASH_FUNC_STADTX  
-DPERL_HASH_FUNC_ZAPHOD32
```

To disable SBOX hashing you can use

```
-DPERL_HASH_USE_SBOX32_ALSO=0
```

And to set the maximum length to use SBOX32 hashing on with:

```
-DSBOX32_MAX_LEN=16
```

The maximum length allowed is 256. There probably isn't much point in setting it higher than the default.

Incompatible Changes

Subroutine attribute and signature order

The experimental subroutine signatures feature has been changed so that subroutine attributes must now come before the signature rather than after. This is because attributes like ":lvalue" can affect the compilation of code within the signature, for example:

```
sub f :lvalue ($a = do { $x = "abc"; return substr($x,0,1) }) { ... }
```

Note that this the second time they have been flipped:

```
sub f :lvalue ($a, $b) { ... }; # 5.20; 5.28 onwards
sub f ($a, $b) :lvalue { ... }; # 5.22 - 5.26
```

Comma-less variable lists in formats are no longer allowed

Omitting the commas between variables passed to formats is no longer allowed. This has been deprecated since Perl 5.000.

The ":locked" and ":unique" attributes have been removed

These have been no-ops and deprecated since Perl 5.12 and 5.10, respectively.

"\N{" with nothing between the braces is now illegal

This has been deprecated since Perl 5.24.

Opening the same symbol as both a file and directory handle is no longer allowed

Using "open()" and "opendir()" to associate both a filehandle and a dirhandle to the same symbol (glob or scalar) has been deprecated since Perl 5.10.

Use of bare "<<" to mean "<<'''" is no longer allowed

Use of a bare terminator has been deprecated since Perl 5.000.

Setting \$/ to a reference to a non-positive integer no longer allowed

This used to work like setting it to "undef", but has been deprecated since Perl 5.20.

Unicode code points with values exceeding "IV_MAX" are now fatal

This was deprecated since Perl 5.24.

The "B::OP::terse" method has been removed

Use "B::Concise::b_terse" instead.

Use of inherited AUTOLOAD for non-methods is no longer allowed

This was deprecated in Perl 5.004.

Use of strings with code points over 0xFF is not allowed for bitwise string operators

Code points over 0xFF do not make sense for bitwise operators and such an operation will now croak, except for a few remaining cases. See perldeprecation.

This was deprecated in Perl 5.24.

Setting "\${^ENCODING}" to a defined value is now illegal

This has been deprecated since Perl 5.22 and a no-op since Perl 5.26.

Backslash no longer escapes colon in PATH for the "-S" switch

Previously the "-S" switch incorrectly treated backslash ("\") as an escape for colon when traversing the "PATH" environment variable. [GH #15584] <<https://github.com/Perl/perl5/issues/15584>>

the -DH (DEBUG_H) misfeature has been removed

On a perl built with debugging support, the "H" flag to the "-D" debugging option has been removed. This was supposed to dump hash values, but has been broken for many years.

Yada-yada is now strictly a statement

By the time of its initial stable release in Perl 5.12, the "..." (yada-yada) operator was explicitly intended to serve as a statement, not an expression. However, the original implementation was confused on this point, leading to inconsistent parsing. The operator was accidentally accepted in a few situations where it did not serve as a complete statement, such as

```
... . "foo";  
... if $a < $b;
```

The parsing has now been made consistent, permitting yada-yada only as a statement. Affected code can use "do{...}" to put a yada-yada into an arbitrary expression context.

Sort algorithm can no longer be specified

Since Perl 5.8, the sort pragma has had subpragmata "_mergesort", "_quicksort", and "_qsort" that can be used to specify which algorithm perl should use to implement the sort builtin. This was always considered a dubious feature that might not last, hence the underscore spellings, and they were documented as not being portable beyond Perl 5.8. These subpragmata have now been deleted, and any attempt to use them is an error. The sort pragma otherwise remains, and the algorithm-neutral "stable" subpragma can be used to control sorting behaviour. [GH #13234] <<https://github.com/Perl/perl5/issues/13234>>

Over-radix digits in floating point literals

Octal and binary floating point literals used to permit any hexadecimal digit to appear after the radix point. The digits are now restricted to those appropriate for the radix, as digits before the radix point always were.

Return type of "unpackstring()"

The return types of the C API functions "unpackstring()" and "unpack_str()" have changed from "I32" to "SSize_t", in order to accommodate datasets of more than two billion items.

Deprecations

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

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

Some uses of unescaped "{" in regexes are no longer fatal

Perl 5.26.0 fatalized some uses of an unescaped left brace, but an exception was made at the last minute, specifically crafted to be a minimal change to allow GNU Autoconf to work. That tool is heavily depended upon, and continues to use the deprecated usage. Its use of an unescaped left brace is one where we have no intention of repurposing "{" to be something other than itself.

That exception is now generalized to include various other such cases where the "{" will not be repurposed.

Note that these uses continue to raise a deprecation message.

Use of unescaped "{" immediately after a "(" in regular expression patterns is deprecated

Using unescaped left braces is officially deprecated everywhere, but it is not enforced in contexts where their use does not interfere with expected extensions to the language. A deprecation is added in this release when the brace appears immediately after an opening parenthesis. Before this, even if the brace was part of a legal quantifier, it was not interpreted as such, but as the literal characters, unlike other quantifiers that follow a "(" which are considered errors. Now, their use will raise a deprecation message, unless turned off.

Assignment to \$[will be fatal in Perl 5.30

Assigning a non-zero value to \$[has been deprecated since Perl 5.12, but was never given a deadline for removal. This has now been scheduled for Perl 5.30.

hostname() won't accept arguments in Perl 5.32

Passing arguments to "Sys::Hostname::hostname()" was already deprecated, but didn't have a removal date. This has now been scheduled for Perl 5.32. [GH #14662]
<<https://github.com/Perl/perl5/issues/14662>>

Module removals

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

Locale::Codes and its associated Country, Currency and Language modules

Performance Enhancements

- ⊕ The start up overhead for creating regular expression patterns with Unicode properties ("`\p{...}`") has been greatly reduced in most cases.
- ⊕ Many string concatenation expressions are now considerably faster, due to the introduction internally of a "multiconcat" opcode which combines multiple concatenations, and optionally a "=" or " .= ", into a single action. For example, apart from retrieving \$s, \$a and \$b, this whole expression is now handled as a single op:

```
$s .= "a=$a b=$b\n"
```

As a special case, if the LHS of an assignment is a lexical variable or "my \$s", the op itself handles retrieving the lexical variable, which is faster.

In general, the more the expression includes a mix of constant strings and variable expressions, the longer the expression, and the more it mixes together non-utf8 and utf8 strings, the more marked the performance improvement. For example on a "x86_64" system, this code has been benchmarked running four times faster:

```
my $s;
my $a = "ab\x{100}cde";
my $b = "fghij";
my $c = "\x{101}klmn";

for my $i (1..10_000_000) {
    $s = "\x{100}wxyz";
    $s .= "foo=$a bar=$b baz=$c";
}
```

In addition, "sprintf" expressions which have a constant format containing only %s and "%%" format elements, and which have a fixed number of arguments, are now also optimised into a "multiconcat" op.

- ⊕ The "ref()" builtin is now much faster in boolean context, since it no longer bothers to construct a

temporary string like "Foo=ARRAY(0x134af48)".

- ⊕ "keys()" in void and scalar contexts is now more efficient.
- ⊕ The common idiom of comparing the result of **index()** with -1 is now specifically optimised, e.g.

```
if (index(...) != -1) { ... }
```
- ⊕ "for()" loops and similar constructs are now more efficient in most cases.
- ⊕ File::Glob has been modified to remove unnecessary backtracking and recursion, thanks to Russ Cox. See <<https://research.swtch.com/glob>> for more details.
- ⊕ The XS-level "SvTRUE()" API function is now more efficient.
- ⊕ Various integer-returning ops are now more efficient in scalar/boolean context.
- ⊕ Slightly improved performance when parsing stash names. [GH #15689]
<<https://github.com/Perl/perl5/issues/15689>>
- ⊕ Calls to "require" for an already loaded module are now slightly faster. [GH #16175]
<<https://github.com/Perl/perl5/issues/16175>>
- ⊕ The performance of pattern matching "[[:ascii:]]" and "[[:^ascii:]]" has been improved significantly except on EBCDIC platforms.
- ⊕ Various optimizations have been applied to matching regular expression patterns, so under the right circumstances, significant performance gains may be noticed. But in an application with many varied patterns, little overall improvement likely will be seen.
- ⊕ Other optimizations have been applied to UTF-8 handling, but these are not typically a major factor in most applications.

Modules and Pragmata

Key highlights in this release across several modules:

Removal of use vars

The usage of "use vars" has been discouraged since the introduction of "our" in Perl 5.6.0. Where possible the usage of this pragma has now been removed from the Perl source code.

This had a slight effect (for the better) on the output of `WARNING_BITS` in `B::Deparse`.

Use of DynaLoader changed to XSLoader in many modules

XSLoader is more modern, and most modules already require perl 5.6 or greater, so no functionality is lost by switching. In some cases, we have also made changes to the local implementation that may not be reflected in the version on CPAN due to a desire to maintain more backwards compatibility.

Updated Modules and Pragmata

- ⊕ `Archive::Tar` has been upgraded from version 2.24 to 2.30.

This update also handled CVE-2018-12015: directory traversal vulnerability. [cpan #125523]
<<https://rt.cpan.org/Ticket/Display.html?id=125523>>

- ⊕ `arybase` has been upgraded from version 0.12 to 0.15.
- ⊕ `Attribute::Handlers` has been upgraded from version 0.99 to 1.01.
- ⊕ `attributes` has been upgraded from version 0.29 to 0.33.
- ⊕ `B` has been upgraded from version 1.68 to 1.74.
- ⊕ `B::Concise` has been upgraded from version 0.999 to 1.003.
- ⊕ `B::Debug` has been upgraded from version 1.24 to 1.26.

NOTE: `B::Debug` is deprecated and may be removed from a future version of Perl.

- ⊕ `B::Deparse` has been upgraded from version 1.40 to 1.48.

It includes many bug fixes, and in particular, it now deparses variable attributes correctly:

```
my $x :foo; # used to deparse as
    # 'attributes'->import('main', \ $x, 'foo'), my $x;
```

- ⊕ `base` has been upgraded from version 2.25 to 2.27.
- ⊕ `bignum` has been upgraded from version 0.47 to 0.49.
- ⊕ `blib` has been upgraded from version 1.06 to 1.07.

- ⊕ bytes has been upgraded from version 1.05 to 1.06.
- ⊕ Carp has been upgraded from version 1.42 to 1.50.

If a package on the call stack contains a constant named "ISA", Carp no longer throws a "Not a GLOB reference" error.

Carp, when generating stack traces, now attempts to work around longstanding bugs resulting from Perl's non-reference-counted stack. [GH #9282]

<<https://github.com/Perl/perl5/issues/9282>>

Carp has been modified to avoid assuming that objects cannot be overloaded without the overload module loaded (this can happen with objects created by XS modules). Previously, infinite recursion would result if an XS-defined overload method itself called Carp. [GH #16407]

<<https://github.com/Perl/perl5/issues/16407>>

Carp now avoids using "overload::StrVal", partly because older versions of overload (included with perl 5.14 and earlier) load Scalar::Util at run time, which will fail if Carp has been invoked after a syntax error.

- ⊕ charnames has been upgraded from version 1.44 to 1.45.
- ⊕ Compress::Raw::Zlib has been upgraded from version 2.074 to 2.076.

This addresses a security vulnerability in older versions of the 'zlib' library (which is bundled with Compress-Raw-Zlib).

- ⊕ Config::Extensions has been upgraded from version 0.01 to 0.02.
- ⊕ Config::Perl::V has been upgraded from version 0.28 to 0.29.
- ⊕ CPAN has been upgraded from version 2.18 to 2.20.
- ⊕ Data::Dumper has been upgraded from version 2.167 to 2.170.

Quoting of glob names now obeys the Useqq option [GH #13274]

<<https://github.com/Perl/perl5/issues/13274>>.

Attempts to set an option to "undef" through a combined getter/setter method are no longer mistaken for getter calls [GH #12135] <<https://github.com/Perl/perl5/issues/12135>>.

- ⊕ Devel::Peek has been upgraded from version 1.26 to 1.27.
- ⊕ Devel::PPPort has been upgraded from version 3.35 to 3.40.

Devel::PPPort has moved from cpan-first to perl-first maintenance

Primary responsibility for the code in Devel::PPPort has moved into core perl. In a practical sense there should be no change except that hopefully it will stay more up to date with changes made to symbols in perl, rather than needing to be updated after the fact.

- ⊕ Digest::SHA has been upgraded from version 5.96 to 6.01.
- ⊕ DirHandle has been upgraded from version 1.04 to 1.05.
- ⊕ DynaLoader has been upgraded from version 1.42 to 1.45.

Its documentation now shows the use of "__PACKAGE__" and direct object syntax [GH #16190]
<<https://github.com/Perl/perl5/issues/16190>>.

- ⊕ Encode has been upgraded from version 2.88 to 2.97.
- ⊕ encoding has been upgraded from version 2.19 to 2.22.
- ⊕ Errno has been upgraded from version 1.28 to 1.29.
- ⊕ experimental has been upgraded from version 0.016 to 0.019.
- ⊕ Exporter has been upgraded from version 5.72 to 5.73.
- ⊕ ExtUtils::CBuilder has been upgraded from version 0.280225 to 0.280230.
- ⊕ ExtUtils::Constant has been upgraded from version 0.23 to 0.25.
- ⊕ ExtUtils::Embed has been upgraded from version 1.34 to 1.35.
- ⊕ ExtUtils::Install has been upgraded from version 2.04 to 2.14.
- ⊕ ExtUtils::MakeMaker has been upgraded from version 7.24 to 7.34.
- ⊕ ExtUtils::Miniperl has been upgraded from version 1.06 to 1.08.

- ⊕ ExtUtils::ParseXS has been upgraded from version 3.34 to 3.39.
- ⊕ ExtUtils::Typemaps has been upgraded from version 3.34 to 3.38.
- ⊕ ExtUtils::XSSymSet has been upgraded from version 1.3 to 1.4.
- ⊕ feature has been upgraded from version 1.47 to 1.52.
- ⊕ fields has been upgraded from version 2.23 to 2.24.
- ⊕ File::Copy has been upgraded from version 2.32 to 2.33.

It will now use the sub-second precision variant of **utime()** supplied by Time::HiRes where available. [GH #16225] <<https://github.com/Perl/perl5/issues/16225>>.

- ⊕ File::Fetch has been upgraded from version 0.52 to 0.56.
- ⊕ File::Glob has been upgraded from version 1.28 to 1.31.
- ⊕ File::Path has been upgraded from version 2.12_01 to 2.15.
- ⊕ File::Spec and Cwd have been upgraded from version 3.67 to 3.74.
- ⊕ File::stat has been upgraded from version 1.07 to 1.08.
- ⊕ FileCache has been upgraded from version 1.09 to 1.10.
- ⊕ Filter::Simple has been upgraded from version 0.93 to 0.95.
- ⊕ Filter::Util::Call has been upgraded from version 1.55 to 1.58.
- ⊕ GDBM_File has been upgraded from version 1.15 to 1.17.

Its documentation now explains that "each" and "delete" don't mix in hashes tied to this module [GH #12894] <<https://github.com/Perl/perl5/issues/12894>>.

It will now retry opening with an acceptable block size if asking gdbm to default the block size failed [GH #13232] <<https://github.com/Perl/perl5/issues/13232>>.

- ⊕ Getopt::Long has been upgraded from version 2.49 to 2.5.

- ⊕ Hash::Util::FieldHash has been upgraded from version 1.19 to 1.20.
- ⊕ I18N::Langinfo has been upgraded from version 0.13 to 0.17.

This module is now available on all platforms, emulating the system **nl_langinfo**(3) on systems that lack it. Some caveats apply, as detailed in its documentation, the most severe being that, except for MS Windows, the "CODESET" item is not implemented on those systems, always returning "".

It now sets the UTF-8 flag in its returned scalar if the string contains legal non-ASCII UTF-8, and the locale is UTF-8 [GH #15131] <<https://github.com/Perl/perl5/issues/15131>>.

This update also fixes a bug in which the underlying locale was ignored for the "RADIXCHAR" (always was returned as a dot) and the "THOUSEP" (always empty). Now the locale-appropriate values are returned.

- ⊕ I18N::LangTags has been upgraded from version 0.42 to 0.43.
- ⊕ if has been upgraded from version 0.0606 to 0.0608.
- ⊕ IO has been upgraded from version 1.38 to 1.39.
- ⊕ IO::Socket::IP has been upgraded from version 0.38 to 0.39.
- ⊕ IPC::Cmd has been upgraded from version 0.96 to 1.00.
- ⊕ JSON::PP has been upgraded from version 2.27400_02 to 2.97001.
- ⊕ The "libnet" distribution has been upgraded from version 3.10 to 3.11.
- ⊕ List::Util has been upgraded from version 1.46_02 to 1.49.
- ⊕ Locale::Codes has been upgraded from version 3.42 to 3.56.

NOTE: Locale::Codes scheduled to be removed from core in Perl 5.30.

- ⊕ Locale::Maketext has been upgraded from version 1.28 to 1.29.
- ⊕ Math::BigInt has been upgraded from version 1.999806 to 1.999811.

- ⊕ `Math::BigInt::FastCalc` has been upgraded from version 0.5005 to 0.5006.
- ⊕ `Math::BigRat` has been upgraded from version 0.2611 to 0.2613.
- ⊕ `Module::CoreList` has been upgraded from version 5.20170530 to 5.20180622.
- ⊕ `mro` has been upgraded from version 1.20 to 1.22.
- ⊕ `Net::Ping` has been upgraded from version 2.55 to 2.62.
- ⊕ `NEXT` has been upgraded from version 0.67 to 0.67_01.
- ⊕ `ODBM_File` has been upgraded from version 1.14 to 1.15.
- ⊕ `Opcodes` has been upgraded from version 1.39 to 1.43.
- ⊕ `overload` has been upgraded from version 1.28 to 1.30.
- ⊕ `PerlIO::encoding` has been upgraded from version 0.25 to 0.26.
- ⊕ `PerlIO::scalar` has been upgraded from version 0.26 to 0.29.
- ⊕ `PerlIO::via` has been upgraded from version 0.16 to 0.17.
- ⊕ `Pod::Functions` has been upgraded from version 1.11 to 1.13.
- ⊕ `Pod::Html` has been upgraded from version 1.2202 to 1.24.

A title for the HTML document will now be automatically generated by default from a "NAME" section in the POD document, as it used to be before the module was rewritten to use `Pod::Simple::XHTML` to do the core of its job [GH #11954]
<<https://github.com/Perl/perl5/issues/11954>>.

- ⊕ `Pod::Perldoc` has been upgraded from version 3.28 to 3.2801.
- ⊕ The "podlators" distribution has been upgraded from version 4.09 to 4.10.

Man page references and function names now follow the Linux man page formatting standards, instead of the Solaris standard.

- ⊕ POSIX has been upgraded from version 1.76 to 1.84.

Some more cautions were added about using locale-specific functions in threaded applications.

- ⊕ re has been upgraded from version 0.34 to 0.36.
- ⊕ Scalar::Util has been upgraded from version 1.46_02 to 1.50.
- ⊕ SelfLoader has been upgraded from version 1.23 to 1.25.
- ⊕ Socket has been upgraded from version 2.020_03 to 2.027.
- ⊕ sort has been upgraded from version 2.02 to 2.04.
- ⊕ Storable has been upgraded from version 2.62 to 3.08.
- ⊕ Sub::Util has been upgraded from version 1.48 to 1.49.
- ⊕ subs has been upgraded from version 1.02 to 1.03.
- ⊕ Sys::Hostname has been upgraded from version 1.20 to 1.22.
- ⊕ Term::ReadLine has been upgraded from version 1.16 to 1.17.
- ⊕ Test has been upgraded from version 1.30 to 1.31.
- ⊕ Test::Harness has been upgraded from version 3.38 to 3.42.
- ⊕ Test::Simple has been upgraded from version 1.302073 to 1.302133.
- ⊕ threads has been upgraded from version 2.15 to 2.22.

The documentation now better describes the problems that arise when returning values from threads, and no longer warns about creating threads in "BEGIN" blocks. [GH #11563]

<<https://github.com/Perl/perl5/issues/11563>>

- ⊕ threads::shared has been upgraded from version 1.56 to 1.58.
- ⊕ Tie::Array has been upgraded from version 1.06 to 1.07.

- ⊕ Tie::StdHandle has been upgraded from version 4.4 to 4.5.
- ⊕ Time::gmtime has been upgraded from version 1.03 to 1.04.
- ⊕ Time::HiRes has been upgraded from version 1.9741 to 1.9759.
- ⊕ Time::localtime has been upgraded from version 1.02 to 1.03.
- ⊕ Time::Piece has been upgraded from version 1.31 to 1.3204.
- ⊕ Unicode::Collate has been upgraded from version 1.19 to 1.25.
- ⊕ Unicode::Normalize has been upgraded from version 1.25 to 1.26.
- ⊕ Unicode::UCD has been upgraded from version 0.68 to 0.70.

The function "num" now accepts an optional parameter to help in diagnosing error returns.

- ⊕ User::grent has been upgraded from version 1.01 to 1.02.
- ⊕ User::pwent has been upgraded from version 1.00 to 1.01.
- ⊕ utf8 has been upgraded from version 1.19 to 1.21.
- ⊕ vars has been upgraded from version 1.03 to 1.04.
- ⊕ version has been upgraded from version 0.9917 to 0.9923.
- ⊕ VMS::DCLsym has been upgraded from version 1.08 to 1.09.
- ⊕ VMS::Stdio has been upgraded from version 2.41 to 2.44.
- ⊕ warnings has been upgraded from version 1.37 to 1.42.

It now includes new functions with names ending in "_at_level", allowing callers to specify the exact call frame. [GH #16257] <<https://github.com/Perl/perl5/issues/16257>>

- ⊕ XS::Typemap has been upgraded from version 0.15 to 0.16.
- ⊕ XSLoader has been upgraded from version 0.27 to 0.30.

Its documentation now shows the use of "`__PACKAGE__`", and direct object syntax for example "DynaLoader" usage [GH #16190] <<https://github.com/Perl/perl5/issues/16190>>.

Platforms that use "mod2fname" to edit the names of loadable libraries now look for bootstrap (.bs) files under the correct, non-edited name.

Removed Modules and Pragmata

- ⊕ The "VMS::stdio" compatibility shim has been removed.

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>>.

Additionally, the following selected changes have been made:

perlapi

- ⊕ The API functions "`perl_parse()`", "`perl_run()`", and "`perl_destruct()`" are now documented comprehensively, where previously the only documentation was a reference to the `perlembed` tutorial.
- ⊕ The documentation of "`newGIVENOP()`" has been belatedly updated to account for the removal of lexical `$_`.
- ⊕ The API functions "`newCONSTSUB()`" and "`newCONSTSUB_flags()`" are documented much more comprehensively than before.

perldata

- ⊕ The section "Truth and Falsehood" in `perlsyn` has been moved into `perldata`.

perldebbugs

- ⊕ The description of the conditions under which "`DB::sub()`" will be called has been clarified. [GH #16055] <<https://github.com/Perl/perl5/issues/16055>>

perldiag

- ⊕ "Variable length lookbehind not implemented in regex `m/%s/`" in `perldiag`

This now gives more ideas as to workarounds to the issue that was introduced in Perl 5.18 (but not documented explicitly in its `perldelta`) for the fact that some Unicode `/i` rules cause a few sequences such as

`(?<!st)`

to be considered variable length, and hence disallowed.

- ⊕ "Use of state `$_` is experimental" in `perldiag`

This entry has been removed, as the experimental support of this construct was removed in perl 5.24.0.

- ⊕ The diagnostic "Initialization of state variables in list context currently forbidden" has changed to "Initialization of state variables in list currently forbidden", because list-context initialization of single aggregate state variables is now permitted.

perlembded

- ⊕ The examples in `perlembded` have been made more portable in the way they exit, and the example that gets an exit code from the embedded Perl interpreter now gets it from the right place. The examples that pass a constructed `argv` to Perl now show the mandatory null `argv[argc]`.
- ⊕ An example in `perlembded` used the string value of `"ERRSV"` as a format string when calling `croak()`. If that string contains format codes such as `%s` this could crash the program.

This has been changed to a call to `croak_sv()`.

An alternative could have been to supply a trivial format string:

```
croak("%s", SvPV_nolen(ERRSV));
```

or as a special case for `"ERRSV"` simply:

```
croak(NULL);
```

perlfunc

- ⊕ There is now a note that warnings generated by built-in functions are documented in `perldiag` and `warnings`. [GH #12642] <<https://github.com/Perl/perl5/issues/12642>>

- ⊕ The documentation for the "exists" operator no longer says that autovivification behaviour "may be fixed in a future release". We've determined that we're not going to change the default behaviour. [GH #15231] <<https://github.com/Perl/perl5/issues/15231>>
- ⊕ A couple of small details in the documentation for the "bless" operator have been clarified. [GH #14684] <<https://github.com/Perl/perl5/issues/14684>>
- ⊕ The description of @INC hooks in the documentation for "require" has been corrected to say that filter subroutines receive a useless first argument. [GH #12569] <<https://github.com/Perl/perl5/issues/12569>>
- ⊕ The documentation of "ref" has been rewritten for clarity.
- ⊕ The documentation of "use" now explains what syntactically qualifies as a version number for its module version checking feature.
- ⊕ The documentation of "warn" has been updated to reflect that since Perl 5.14 it has treated complex exception objects in a manner equivalent to "die". [GH #13641] <<https://github.com/Perl/perl5/issues/13641>>
- ⊕ The documentation of "die" and "warn" has been revised for clarity.
- ⊕ The documentation of "each" has been improved, with a slightly more explicit description of the sharing of iterator state, and with caveats regarding the fragility of while-each loops. [GH #16334] <<https://github.com/Perl/perl5/issues/16334>>
- ⊕ Clarification to "require" was added to explain the differences between

```
require Foo::Bar;  
require "Foo/Bar.pm";
```

perlgit

- ⊕ The precise rules for identifying "smoke-me" branches are now stated.

perlguits

- ⊕ The section on reference counting in perlguits has been heavily revised, to describe references in the way a programmer needs to think about them rather than in terms of the physical data structures.

- ⊕ Improve documentation related to UTF-8 multibytes.

perlintern

- ⊕ The internal functions "newXS_len_flags()" and "newATTRSUB_x()" are now documented.

perlobj

- ⊕ The documentation about "DESTROY" methods has been corrected, updated, and revised, especially in regard to how they interact with exceptions. [GH #14083]
<<https://github.com/Perl/perl5/issues/14083>>

perlop

- ⊕ The description of the "x" operator in perlop has been clarified. [GH #16253]
<<https://github.com/Perl/perl5/issues/16253>>
- ⊕ perlop has been updated to note that "qw"'s whitespace rules differ from that of "split"'s in that only ASCII whitespace is used.
- ⊕ The general explanation of operator precedence and associativity has been corrected and clarified. [GH #15153] <<https://github.com/Perl/perl5/issues/15153>>
- ⊕ The documentation for the "\" referencing operator now explains the unusual context that it supplies to its operand. [GH #15932] <<https://github.com/Perl/perl5/issues/15932>>

perlrequick

- ⊕ Clarifications on metacharacters and character classes

perlretut

- ⊕ Clarify metacharacters.

perlrun

- ⊕ Clarify the differences between **-M** and **-m**. [GH #15998]
<<https://github.com/Perl/perl5/issues/15998>>

perlsec

- ⊕ The documentation about set-id scripts has been updated and revised. [GH #10289]
<<https://github.com/Perl/perl5/issues/10289>>

- ⊕ A section about using "sudo" to run Perl scripts has been added.

perlsyn

- ⊕ The section "Truth and Falsehood" in perlsyn has been removed from that document, where it didn't belong, and merged into the existing paragraph on the same topic in perldata.
- ⊕ The means to disambiguate between code blocks and hash constructors, already documented in perlref, are now documented in perlsyn too. [GH #15918]
<<https://github.com/Perl/perl5/issues/15918>>

perluniprops

- ⊕ perluniprops has been updated to note that "\p{Word}" now includes code points matching the "\p{Join_Control}" property. The change to the property was made in Perl 5.18, but not documented until now. There are currently only two code points that match this property U+200C (ZERO WIDTH NON-JOINER) and U+200D (ZERO WIDTH JOINER).
- ⊕ For each binary table or property, the documentation now includes which characters in the range "\x00-\xFF" it matches, as well as a list of the first few ranges of code points matched above that.

perlvar

- ⊕ The entry for \$+ in perlvar has been expanded upon to describe handling of multiply-named capturing groups.

perlfunc, perlop, perlsyn

- ⊕ In various places, improve the documentation of the special cases in the condition expression of a while loop, such as implicit "defined" and assignment to \$_. [GH #16334]
<<https://github.com/Perl/perl5/issues/16334>>

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

- ⊕ Can't "goto" into a "given" block

(F) A "goto" statement was executed to jump into the middle of a "given" block. You can't get there from here. See "goto" in perlfunc.

- ⊕ Can't "goto" into a binary or list expression

Use of "goto" to jump into the parameter of a binary or list operator has been prohibited, to prevent crashes and stack corruption. [GH #15914] <<https://github.com/Perl/perl5/issues/15914>>

You may only enter the *first* argument of an operator that takes a fixed number of arguments, since this is a case that will not cause stack corruption. [GH #16415] <<https://github.com/Perl/perl5/issues/16415>>

New Warnings

- ⊕ Old package separator used in string

(W syntax) You used the old package separator, "", in a variable named inside a double-quoted string; e.g., "In \$name's house". This is equivalent to "In \$name::s house". If you meant the former, put a backslash before the apostrophe ("In \$name\'s house").

- ⊕ "Locale '%s' contains (at least) the following characters which have unexpected meanings: %s The Perl program will use the expected meanings" in perldiag

Changes to Existing Diagnostics

- ⊕ A false-positive warning that was issued when using a numerically-quantified sub-pattern in a recursive regex has been silenced. [GH #16106] <<https://github.com/Perl/perl5/issues/16106>>
- ⊕ The warning about useless use of a concatenation operator in void context is now generated for expressions with multiple concatenations, such as "\$a.\$b.\$c", which used to mistakenly not warn. [GH #3990] <<https://github.com/Perl/perl5/issues/3990>>
- ⊕ Warnings that a variable or subroutine "masks earlier declaration in same ...", or that an "our" variable has been redeclared, have been moved to a new warnings category "shadow". Previously they were in category "misc".
- ⊕ The deprecation warning from "Sys::Hostname::hostname()" saying that it doesn't accept

arguments now states the Perl version in which the warning will be upgraded to an error. [GH #14662] <<https://github.com/Perl/perl5/issues/14662>>

- ⊕ The `perldiag` entry for the error regarding a `set-id` script has been expanded to make clear that the error is reporting a specific security vulnerability, and to advise how to fix it.
- ⊕ The "Unable to flush stdout" error message was missing a trailing newline. [debian #875361]

Utility Changes

perlbug

- ⊕ "--help" and "--version" options have been added.

Configuration and Compilation

- ⊕ C89 requirement

Perl has been documented as requiring a C89 compiler to build since October 1998. A variety of simplifications have now been made to Perl's internals to rely on the features specified by the C89 standard. We believe that this internal change hasn't altered the set of platforms that Perl builds on, but please report a bug if Perl now has new problems building on your platform.

- ⊕ On GCC, "-Werror=pointer-arith" is now enabled by default, disallowing arithmetic on void and function pointers.
- ⊕ Where an HTML version of the documentation is installed, the HTML documents now use relative links to refer to each other. Links from the index page of `perlipc` to the individual section documents are now correct. [GH #11941] <<https://github.com/Perl/perl5/issues/11941>>
- ⊕ *lib/unicore/mktables* now correctly canonicalizes the names of the dependencies stored in the files it generates.

regen/mk_invlists.pl, unlike the other *regen/*.pl* scripts, used `$0` to name itself in the dependencies stored in the files it generates. It now uses a literal so that the path stored in the generated files doesn't depend on how *regen/mk_invlists.pl* is invoked.

This lack of canonical names could cause test failures in *t/porting/regen.t*. [GH #16446] <<https://github.com/Perl/perl5/issues/16446>>

- ⊕ New probes

`HAS_BUILTIN_ADD_OVERFLOW`

HAS_BUILTIN_MUL_OVERFLOW
HAS_BUILTIN_SUB_OVERFLOW
HAS_THREAD_SAFE_NL_LANGINFO_L
HAS_LOCALECONV_L
HAS_MBLEN
HAS_MBRTOWC
HAS_MEMRCHR
HAS_NANOSLEEP
HAS_STRNLEN
HAS_STRTOLD_L
I_WCHAR

Testing

- ⊕ Testing of the XS-APItest directory is now done in parallel, where applicable.
- ⊕ Perl now includes a default *.travis.yml* file for Travis CI testing on github mirrors. [GH #14558]
[<https://github.com/Perl/perl5/issues/14558>](https://github.com/Perl/perl5/issues/14558)
- ⊕ The watchdog timer count in *re/pat_pscho.t* can now be overridden.

This test can take a long time to run, so there is a timer to keep this in check (currently, 5 minutes). This commit adds checking the environment variable "PERL_TEST_TIME_OUT_FACTOR"; if set, the time out setting is multiplied by its value.

- ⊕ *harness* no longer waits for 30 seconds when running *t/io/openpid.t*. [GH #13535]
[<https://github.com/Perl/perl5/issues/13535>](https://github.com/Perl/perl5/issues/13535) [GH #16420]
[<https://github.com/Perl/perl5/issues/16420>](https://github.com/Perl/perl5/issues/16420)

Packaging

For the past few years we have released perl using three different archive formats: bzip (".bz2"), LZMA2 (".xz") and gzip (".gz"). Since xz compresses better and decompresses faster, and gzip is more compatible and uses less memory, we have dropped the ".bz2" archive format with this release. (If this poses a problem, do let us know; see "Reporting Bugs", below.)

Platform Support

Discontinued Platforms

PowerUX / Power MAX OS

Compiler hints and other support for these apparently long-defunct platforms has been removed.

Platform-Specific Notes

CentOS

Compilation on CentOS 5 is now fixed.

Cygwin

A build with the quadmath library can now be done on Cygwin.

Darwin

Perl now correctly uses reentrant functions, like "asctime_r", on versions of Darwin that have support for them.

FreeBSD

FreeBSD's */usr/share/mk/sys.mk* specifies "-O2" for architectures other than ARM and MIPS. By default, perl is now compiled with the same optimization levels.

VMS

Several fix-ups for *configure.com*, marking function VMS has (or doesn't have).

CRTL features can now be set by embedders before invoking Perl by using the "decc\$feature_set" and "decc\$feature_set_value" functions. Previously any attempt to set features after image initialization were ignored.

Windows

- ⊕ Support for compiling perl on Windows using Microsoft Visual Studio 2017 (containing Visual C

14.1) has been added.

- ⊕ Visual C

compiler version detection has been improved to work on non-English language systems.

- ⊕ We now set \$Config{libpth} correctly for 64-bit builds using Visual C

versions earlier than 14.1.

Internal Changes

- ⊕ A new optimisation phase has been added to the compiler, "optimize_optree()", which does a top-

down scan of a complete optree just before the peephole optimiser is run. This phase is not currently hookable.

- ⊕ An "OP_MULTICONCAT" op has been added. At "optimize_optree()" time, a chain of "OP_CONCAT" and "OP_CONST" ops, together optionally with an "OP_STRINGIFY" and/or "OP_SASSIGN", are combined into a single "OP_MULTICONCAT" op. The op is of type "UNOP_AUX", and the aux array contains the argument count, plus a pointer to a constant string and a set of segment lengths. For example with

```
my $x = "foo=$foo, bar=$bar\n";
```

the constant string would be "foo=, bar=\n" and the segment lengths would be (4,6,1). If the string contains characters such as "\x80", whose representation changes under utf8, two sets of strings plus lengths are precomputed and stored.

- ⊕ Direct access to "PL_keyword_plugin" is not safe in the presence of multithreading. A new "wrap_keyword_plugin" function has been added to allow XS modules to safely define custom keywords even when loaded from a thread, analogous to "PL_check" / "wrap_op_checker".
- ⊕ The "PL_statbuf" interpreter variable has been removed.
- ⊕ The deprecated function "to_utf8_case()", accessible from XS code, has been removed.
- ⊕ A new function "is_utf8_invariant_string_loc()" has been added that is like "is_utf8_invariant_string()" but takes an extra pointer parameter into which is stored the location of the first variant character, if any are found.
- ⊕ A new function, "Perl_langinfo()" has been added. It is an (almost) drop-in replacement for the system nl_langinfo(3), but works on platforms that lack that; as well as being more thread-safe, and hiding some gotchas with locale handling from the caller. Code that uses this, needn't use localeconv(3) (and be affected by the gotchas) to find the decimal point, thousands separator, or currency symbol. See "Perl_langinfo" in perlapi.
- ⊕ A new API function "sv_rvunweaken()" has been added to complement "sv_rvweaken()". The implementation was taken from "unweaken" in Scalar::Util.
- ⊕ A new flag, "SORTf_UNSTABLE", has been added. This will allow a future commit to make mergesort unstable when the user specifies Xno sort stableX, since it has been decided that mergesort should remain stable by default.

- ⊕ XS modules can now automatically get reentrant versions of system functions on threaded perls.

By adding

```
#define PERL_REENTRANT
```

near the beginning of an "XS" file, it will be compiled so that whatever reentrant functions perl knows about on that system will automatically and invisibly be used instead of the plain, non-reentrant versions. For example, if you write "getpwnam()" in your code, on a system that has "getpwnam_r()" all calls to the former will be translated invisibly into the latter. This does not happen except on threaded perls, as they aren't needed otherwise. Be aware that which functions have reentrant versions varies from system to system.

- ⊕ The "PERL_NO_OP_PARENT" build define is no longer supported, which means that perl is now always built with "PERL_OP_PARENT" enabled.
- ⊕ The format and content of the non-utf8 transliteration table attached to the "op_pv" field of "OP_TRANS"/"OP_TRANSR" ops has changed. It's now a "struct OPtrans_map".
- ⊕ A new compiler "#define", "dTHX_DEBUGGING". has been added. This is useful for XS or C code that only need the thread context because their debugging statements that get compiled only under "-DDEBUGGING" need one.
- ⊕ A new API function "Perl_setlocale" in perlapi has been added.
- ⊕ "sync_locale" in perlapi has been revised to return a boolean as to whether the system was using the global locale or not.
- ⊕ A new kind of magic scalar, called a "nonelem" scalar, has been introduced. It is stored in an array to denote a non-existent element, whenever such an element is accessed in a potential lvalue context. It replaces the existing "defelem" (deferred element) magic wherever this is possible, being significantly more efficient. This means that "some_sub(\$sparse_array[\$nonelem])" no longer has to create a new magic defelem scalar each time, as long as the element is within the array.

It partially fixes the rare bug of deferred elements getting out of synch with their arrays when the array is shifted or unshifted. [GH #16364] <<https://github.com/Perl/perl5/issues/16364>>

Selected Bug Fixes

- ⊕ List assignment ("aassign") could in some rare cases allocate an entry on the mortals stack and

leave the entry uninitialized, leading to possible crashes. [GH #16017]
[<https://github.com/Perl/perl5/issues/16017>](https://github.com/Perl/perl5/issues/16017)

- ⊕ Attempting to apply an attribute to an "our" variable where a function of that name already exists could result in a NULL pointer being supplied where an SV was expected, crashing perl. [perl #131597] [<https://rt.perl.org/Ticket/Display.html?id=131597>](https://rt.perl.org/Ticket/Display.html?id=131597)
- ⊕ "split ' '" now correctly handles the argument being split when in the scope of the "unicode_strings" feature. Previously, when a string using the single-byte internal representation contained characters that are whitespace by Unicode rules but not by ASCII rules, it treated those characters as part of fields rather than as field separators. [GH #15904] [<https://github.com/Perl/perl5/issues/15904>](https://github.com/Perl/perl5/issues/15904)
- ⊕ Several built-in functions previously had bugs that could cause them to write to the internal stack without allocating room for the item being written. In rare situations, this could have led to a crash. These bugs have now been fixed, and if any similar bugs are introduced in future, they will be detected automatically in debugging builds.

These internal stack usage checks introduced are also done by the "entersub" operator when calling XSUBs. This means we can report which XSUB failed to allocate enough stack space. [GH #16126] [<https://github.com/Perl/perl5/issues/16126>](https://github.com/Perl/perl5/issues/16126)

- ⊕ Using a symbolic ref with postderef syntax as the key in a hash lookup was yielding an assertion failure on debugging builds. [GH #16029] [<https://github.com/Perl/perl5/issues/16029>](https://github.com/Perl/perl5/issues/16029)
- ⊕ Array and hash variables whose names begin with a caret now admit indexing inside their curlies when interpolated into strings, as in "\${^CAPTURE[0]}" to index "@{^CAPTURE}". [GH #16050] [<https://github.com/Perl/perl5/issues/16050>](https://github.com/Perl/perl5/issues/16050)
- ⊕ Fetching the name of a glob that was previously UTF-8 but wasn't any longer would return that name flagged as UTF-8. [GH #15971] [<https://github.com/Perl/perl5/issues/15971>](https://github.com/Perl/perl5/issues/15971)
- ⊕ The perl "sprintf()" function (via the underlying C function "Perl_sv_vcatpvfn_flags()") has been heavily reworked to fix many minor bugs, including the integer wrapping of large width and precision specifiers and potential buffer overruns. It has also been made faster in many cases.
- ⊕ Exiting from an "eval", whether normally or via an exception, now always frees temporary values (possibly calling destructors) *before* setting \$@. For example:

```
sub DESTROY { eval { die "died in DESTROY"; } }
```

```
eval { bless [], };
```

\$@ used to be equal to "died in DESTROY" here; it's now "".

- ⊕ Fixed a duplicate symbol failure with "-flto -mieee-fp" builds. *pp.c* defined "_LIB_VERSION" which "-lieee" already defines. [GH #16086] <<https://github.com/Perl/perl5/issues/16086>>
- ⊕ The tokenizer no longer consumes the exponent part of a floating point number if it's incomplete. [GH #16073] <<https://github.com/Perl/perl5/issues/16073>>
- ⊕ On non-threaded builds, for "m/\$null/" where \$null is an empty string is no longer treated as if the "/o" flag was present when the previous matching match operator included the "/o" flag. The rewriting used to implement this behavior could confuse the interpreter. This matches the behaviour of threaded builds. [GH #14668] <<https://github.com/Perl/perl5/issues/14668>>
- ⊕ Parsing a "sub" definition could cause a use after free if the "sub" keyword was followed by whitespace including newlines (and comments.) [GH #16097] <<https://github.com/Perl/perl5/issues/16097>>
- ⊕ The tokenizer now correctly adjusts a parse pointer when skipping whitespace in a "\${identifier}" construct. [perl #131949] <<https://rt.perl.org/Public/Bug/Display.html?id=131949>>
- ⊕ Accesses to "\${^LAST_FH}" no longer assert after using any of a variety of I/O operations on a non-glob. [GH #15372] <<https://github.com/Perl/perl5/issues/15372>>
- ⊕ The XS-level "Copy()", "Move()", "Zero()" macros and their variants now assert if the pointers supplied are "NULL". ISO C considers supplying NULL pointers to the functions these macros are built upon as undefined behaviour even when their count parameters are zero. Based on these assertions and the original bug report three macro calls were made conditional. [GH #16079] <<https://github.com/Perl/perl5/issues/16079>> [GH #16112] <<https://github.com/Perl/perl5/issues/16112>>
- ⊕ Only the "=" operator is permitted for defining defaults for parameters in subroutine signatures. Previously other assignment operators, e.g. "+=", were also accidentally permitted. [GH #16084] <<https://github.com/Perl/perl5/issues/16084>>
- ⊕ Package names are now always included in ":prototype" warnings [perl #131833] <<https://rt.perl.org/Public/Bug/Display.html?id=131833>>
- ⊕ The "je_old_stack_hwm" field, previously only found in the "jmpenv" structure on debugging builds, has been added to non-debug builds as well. This fixes an issue with some CPAN modules

caused by the size of this structure varying between debugging and non-debugging builds. [GH #16122] <<https://github.com/Perl/perl5/issues/16122>>

- ⊕ The arguments to the "ninstr()" macro are now correctly parenthesized.
- ⊕ A NULL pointer dereference in the "S_regmatch()" function has been fixed. [perl #132017] <<https://rt.perl.org/Public/Bug/Display.html?id=132017>>
- ⊕ Calling exec PROGRAM LIST with an empty "LIST" has been fixed. This should call "execvp()" with an empty "argv" array (containing only the terminating "NULL" pointer), but was instead just returning false (and not setting \$!). [GH #16075] <<https://github.com/Perl/perl5/issues/16075>>
- ⊕ The "gv_fetchmeth_sv" C function stopped working properly in Perl 5.22 when fetching a constant with a UTF-8 name if that constant subroutine was stored in the stash as a simple scalar reference, rather than a full typeglob. This has been corrected.
- ⊕ Single-letter debugger commands followed by an argument which starts with punctuation (e.g. "p\$^V" and "x@ARGV") now work again. They had been wrongly requiring a space between the command and the argument. [GH #13342] <<https://github.com/Perl/perl5/issues/13342>>
- ⊕ splice now throws an exception ("Modification of a read-only value attempted") when modifying a read-only array. Until now it had been silently modifying the array. The new behaviour is consistent with the behaviour of push and unshift. [GH #15923] <<https://github.com/Perl/perl5/issues/15923>>
- ⊕ "stat()", "lstat()", and file test operators now fail if given a filename containing a nul character, in the same way that "open()" already fails.
- ⊕ "stat()", "lstat()", and file test operators now reliably set \$! when failing due to being applied to a closed or otherwise invalid file handle.
- ⊕ File test operators for Unix permission bits that don't exist on a particular platform, such as "-k" (sticky bit) on Windows, now check that the file being tested exists before returning the blanket false result, and yield the appropriate errors if the argument doesn't refer to a file.
- ⊕ Fixed a 'read before buffer' overrun when parsing a range starting with "\N{" at the beginning of the character set for the transliteration operator. [GH #16189] <<https://github.com/Perl/perl5/issues/16189>>
- ⊕ Fixed a leaked scalar when parsing an empty "\N{" at compile-time. [GH #16189]

<<https://github.com/Perl/perl5/issues/16189>>

- ⊕ Calling "do \$path" on a directory or block device now yields a meaningful error code in \$!. [GH #14841] <<https://github.com/Perl/perl5/issues/14841>>
- ⊕ Regexp substitution using an overloaded replacement value that provides a tainted stringification now correctly taints the resulting string. [GH #12495] <<https://github.com/Perl/perl5/issues/12495>>
- ⊕ Lexical sub declarations in "do" blocks such as "do { my sub lex; 123 }" could corrupt the stack, erasing items already on the stack in the enclosing statement. This has been fixed. [GH #16243] <<https://github.com/Perl/perl5/issues/16243>>
- ⊕ "pack" and "unpack" can now handle repeat counts and lengths that exceed two billion. [GH #13179] <<https://github.com/Perl/perl5/issues/13179>>
- ⊕ Digits past the radix point in octal and binary floating point literals now have the correct weight on platforms where a floating point significand doesn't fit into an integer type.
- ⊕ The canonical truth value no longer has a spurious special meaning as a callable subroutine. It used to be a magic placeholder for a missing "import" or "unimport" method, but is now treated like any other string 1. [GH #14902] <<https://github.com/Perl/perl5/issues/14902>>
- ⊕ "system" now reduces its arguments to strings in the parent process, so any effects of stringifying them (such as overload methods being called or warnings being emitted) are visible in the way the program expects. [GH #13561] <<https://github.com/Perl/perl5/issues/13561>>
- ⊕ The "readpipe()" built-in function now checks at compile time that it has only one parameter expression, and puts it in scalar context, thus ensuring that it doesn't corrupt the stack at runtime. [GH #2793] <<https://github.com/Perl/perl5/issues/2793>>
- ⊕ "sort" now performs correct reference counting when aliasing \$a and \$b, thus avoiding premature destruction and leakage of scalars if they are re-aliased during execution of the sort comparator. [GH #11422] <<https://github.com/Perl/perl5/issues/11422>>
- ⊕ "reverse" with no operand, reversing \$_ by default, is no longer in danger of corrupting the stack. [GH #16291] <<https://github.com/Perl/perl5/issues/16291>>
- ⊕ "exec", "system", et al are no longer liable to have their argument lists corrupted by reentrant calls and by magic such as tied scalars. [GH #15660] <<https://github.com/Perl/perl5/issues/15660>>

- ⊕ Perl's own "malloc" no longer gets confused by attempts to allocate more than a gigabyte on a 64-bit platform. [GH #13273] <<https://github.com/Perl/perl5/issues/13273>>
- ⊕ Stacked file test operators in a sort comparator expression no longer cause a crash. [GH #15626] <<https://github.com/Perl/perl5/issues/15626>>
- ⊕ An identity "tr///" transformation on a reference is no longer mistaken for that reference for the purposes of deciding whether it can be assigned to. [GH #15812] <<https://github.com/Perl/perl5/issues/15812>>
- ⊕ Lengthy hexadecimal, octal, or binary floating point literals no longer cause undefined behaviour when parsing digits that are of such low significance that they can't affect the floating point value. [GH #16114] <<https://github.com/Perl/perl5/issues/16114>>
- ⊕ "open \$\$scalarref..." and similar invocations no longer leak the file handle. [GH #12593] <<https://github.com/Perl/perl5/issues/12593>>
- ⊕ Some convoluted kinds of regexp no longer cause an arithmetic overflow when compiled. [GH #16113] <<https://github.com/Perl/perl5/issues/16113>>
- ⊕ The default typemap, by avoiding "newGVgen", now no longer leaks when XSUBs return file handles ("PerlIO *" or "FILE *"). [GH #12593] <<https://github.com/Perl/perl5/issues/12593>>
- ⊕ Creating a "BEGIN" block as an XS subroutine with a prototype no longer crashes because of the early freeing of the subroutine.
- ⊕ The "printf" format specifier "%.0f" no longer rounds incorrectly [GH #9125] <<https://github.com/Perl/perl5/issues/9125>>, and now shows the correct sign for a negative zero.
- ⊕ Fixed an issue where the error "Scalar value @arrayname[0] better written as \$arrayname" would give an error "Cannot printf Inf with 'c'" when arrayname starts with "Inf". [GH #16335] <<https://github.com/Perl/perl5/issues/16335>>
- ⊕ The Perl implementation of "getcwd()" in "Cwd" in the PathTools distribution now behaves the same as XS implementation on errors: it returns an error, and sets \$!. [GH #16338] <<https://github.com/Perl/perl5/issues/16338>>
- ⊕ Vivify array elements when putting them on the stack. Fixes [GH #5310] <<https://github.com/Perl/perl5/issues/5310>> (reported in April 2002).

- ⊕ Fixed parsing of braced subscript after parens. Fixes [GH #4688] <<https://github.com/Perl/perl5/issues/4688>> (reported in December 2001).
- ⊕ "tr/non_utf8/long_non_utf8/c" could give the wrong results when the length of the replacement character list was greater than 0x7fff.
- ⊕ "tr/non_utf8/non_utf8/cd" failed to add the implied "\x{100}-\x{7fffffff}" to the search character list.
- ⊕ Compilation failures within "perl-within-perl" constructs, such as with string interpolation and the right part of "s///e", now cause compilation to abort earlier.

Previously compilation could continue in order to report other errors, but the failed sub-parse could leave partly parsed constructs on the parser shift-reduce stack, confusing the parser, leading to perl crashes. [GH #14739] <<https://github.com/Perl/perl5/issues/14739>>

- ⊕ On threaded perls where the decimal point (radix) character is not a dot, it has been possible for a race to occur between threads when one needs to use the real radix character (such as with "sprintf"). This has now been fixed by use of a mutex on systems without thread-safe locales, and the problem just doesn't come up on those with thread-safe locales.
- ⊕ Errors while compiling a regex character class could sometime trigger an assertion failure. [GH #16172] <<https://github.com/Perl/perl5/issues/16172>>

Acknowledgements

Perl 5.28.0 represents approximately 13 months of development since Perl 5.26.0 and contains approximately 730,000 lines of changes across 2,200 files from 77 authors.

Excluding auto-generated files, documentation and release tools, there were approximately 580,000 lines of changes to 1,300 .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 5.28.0:

'aBjarmason, Alberto, Simo~es, rAlexandr Savca, Andrew Fresh, Andy Dougherty, Andy Lester, Aristotle Pagaltzis, Ask Bjorn Hansen, Chris 'BinGOs' Williams, Craig A. Berry, Dagfinn Ilmari Mannsaaker, Dan Collins, Daniel Dragan, David Cantrell, David Mitchell, Dmitry Ulanov, Dominic Hargreaves, E. Choroba, Eric Herman, Eugen Konkov, Father Chrysostomos, Gene Sullivan, George Hartzell, Graham Knop, Harald Joerg, H.Merijn Brand, Hugo van der Sanden, Jacques Germishuys, James E Keenan, Jarkko Hietaniemi, Jerry D. Hedden, J. Nick Koston, John Lightsey, John Peacock, John P. Linderman,

John SJ Anderson, Karen Etheridge, Karl Williamson, Ken Brown, Ken Cotterill, Leon Timmermans, Lukas Mai, Marco Fontani, Marc-Philip Werner, Matthew Horsfall, Neil Bowers, Nicholas Clark, Nicolas R., Niko Tyni, Pali, Paul Marquess, Peter John Acklam, Reini Urban, Renee Baecker, Ricardo Signes, Robin Barker, Sawyer X, Scott Lanning, Sergey Aleynikov, Shirakata Kentaro, Shoichi Kaji, Slaven Rezic, Smylers, Steffen Mueller, Steve Hay, Sullivan Beck, Thomas Sibley, Todd Rinaldo, Tomasz Konojacki, Tom Hukins, Tom Wyant, Tony Cook, 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://rt.perl.org/>](https://rt.perl.org/) . There may also be information at [<http://www.perl.org/>](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.