

## NAME

perl5214delta - what is new for perl v5.21.4

## DESCRIPTION

This document describes differences between the 5.21.3 release and the 5.21.4 release.

If you are upgrading from an earlier release such as 5.21.2, first read *perl5213delta*, which describes differences between 5.21.2 and 5.21.3.

## Core Enhancements

### Infinity and NaN (not-a-number) handling improved

Floating point values are able to hold the special values infinity (also -infinity), and NaN (not-a-number). Now we more robustly recognize and propagate the value in computations, and on output normalize them to `Inf` and `NaN`.

See also the *POSIX* enhancements.

## Security

### Perl is now compiled with `-fstack-protector-strong` if available

Perl has been compiled with the anti-stack-smashing option `-fstack-protector` since 5.10.1. Now Perl uses the newer variant called `-fstack-protector-strong`, if available.

## Incompatible Changes

### Changes to the `*` prototype

The `*` character in a subroutine's prototype used to allow barewords to take precedence over most, but not all subroutines. It was never consistent and exhibited buggy behaviour.

Now it has been changed, so subroutines always take precedence over barewords, which brings it into conformity with similarly prototyped built-in functions:

```
sub splat(*) { ... }
sub foo { ... }
splat(foo); # now always splat(foo())
splat(bar); # still splat('bar') as before
close(foo); # close(foo())
close(bar); # close('bar')
```

## Performance Enhancements

- Subroutines with an empty prototype and bodies containing just `undef` are now eligible for inlining. [*perl #122728*]
- Subroutines in packages no longer need to carry typeglobs around with them. Declaring a subroutine will now put a simple sub reference in the stash if possible, saving memory. The typeglobs still notionally exist, so accessing them will cause the subroutine reference to be upgraded to a typeglob. This optimization does not currently apply to XSUBs or exported subroutines, and method calls will undo it, since they cache things in typeglobs. [*perl #120441*]

## Modules and Pragmata

### New Modules and Pragmata

- *B::Op\_private* provides detailed information about the flags used in the `op_private` field of perl opcodes.

### Updated Modules and Pragmata

- *Archive::Tar* has been upgraded from version 2.00 to 2.02.  
Tests can now be run in parallel.

- *Attribute::Handlers* has been upgraded from version 0.96 to 0.97.  
Internal changes to account for the fact that subroutines in packages no longer need to carry typeglobs around with them (see under *Performance Enhancements*).
- *attributes* has been upgraded from version 0.22 to 0.23.  
The usage of `memEQs` in the XS has been corrected. [*perl #122701*]
- *B* has been upgraded from version 1.50 to 1.51.  
It provides a new `B::safename` function, based on the existing `B::GV->SAFENAME`, that converts `"\cOPEN"` to `"^OPEN"`.
- *B::Concise* has been upgraded from version 0.992 to 0.993.  
Internal changes to account for the fact that the defines and labels for the flags in the `op_private` field of OPs are now auto-generated (see under *Internal Changes*).
- *B::Deparse* has been upgraded from version 1.27 to 1.28.  
It now deparses `our(LIST)` and typed lexical (`my Dog $spot`) correctly.
- *bignum* has been upgraded from version 0.37 to 0.38.  
An `eval BLOCK` rather than an `eval EXPR` is now used to see if we can find `Math::BigInt::Lite`.
- *constant* has been upgraded from version 1.31 to 1.32.  
It now accepts fully-qualified constant names, allowing constants to be defined in packages other than the caller.
- *CPAN::Meta::Requirements* has been upgraded from version 2.126 to 2.128.  
Works around limitations in `version::vpp` detecting v-string magic and adds support for forthcoming *ExtUtils::MakeMaker* bootstrap *version.pm* for Perls older than 5.10.0.
- *Data::Dumper* has been upgraded from version 2.152 to 2.154.  
Fixes CVE-2014-4330 by adding a configuration variable/option to limit recursion when dumping deep data structures.
- *experimental* has been upgraded from version 0.008 to 0.010.  
Hardcodes features for Perls older than 5.15.7.
- *ExtUtils::CBuilder* has been upgraded from version 0.280217 to 0.280219.  
Fixes a regression on Android. [*perl #122675*]
- *ExtUtils::Install* has been upgraded from version 1.68 to 2.04.  
No changes to installed files other than version bumps.
- *ExtUtils::Manifest* has been upgraded from version 1.65 to 1.68.  
Fixes a bug with `maniread()`'s handling of quoted filenames and improves `manifind()` to follow symlinks. [*perl #122415*]
- *File::Find* has been upgraded from version 1.27 to 1.28.  
`find()` and `finddepth()` will now warn if passed inappropriate or misspelled options.
- *Getopt::Std* has been upgraded from version 1.10 to 1.11.  
Corrects a typo in the documentation.
- *HTTP::Tiny* has been upgraded from version 0.047 to 0.049.  
`keep_alive` is now fork-safe and thread-safe.

- *IO* has been upgraded from version 1.33 to 1.34.  
The XS implementation has been fixed for the sake of older Perls.
- *IO::Socket::IP* has been upgraded from version 0.31 to 0.32.  
Implements Timeout for `connect()`. [*cpan #92075*]
- *Locale::Codes* has been upgraded from version 3.31 to 3.32.  
New codes have been added.
- *Math::BigInt* has been upgraded from version 1.9996 to 1.9997.  
The documentation now gives test examples using *Test::More* rather than *Test*.
- *Module::CoreList* has been upgraded from version 5.021003 to 5.20140920.  
Updated to cover the latest releases of Perl.
- *overload* has been upgraded from version 1.22 to 1.23.  
A redundant `ref $sub` check has been removed.
- *PathTools* has been upgraded from version 3.49 to 3.50.  
A warning from the **gcc** compiler is now avoided when building the XS.
- *Pod::Perldoc* has been upgraded from version 3.23 to 3.24.  
Filehandles opened for reading or writing now have `:encoding(UTF-8)` set. [*cpan #98019*]
- *POSIX* has been upgraded from version 1.42 to 1.43.  
The C99 math functions and constants (for example `acosh`, `isinf`, `isnan`, `round`, `trunc`; `M_E`, `M_SQRT2`, `M_PI`) have been added.
- *Scalar-List-Utils* has been upgraded from version 1.39 to 1.41.  
A new module, *Sub::Util*, has been added, containing functions related to CODE refs, including `subname` (inspired by *Sub::Identity*) and `set_subname` (copied and renamed from *Sub::Name*).  
The use of `GetMagic` in `List::Util::reduce()` has also been fixed. [*cpan #63211*]
- *Term::ReadLine* has been upgraded from version 1.14 to 1.15.  
Faster checks are now made first in some if-statements.
- *Test::Harness* has been upgraded from version 3.32 to 3.33.  
Various documentation fixes.
- *Test::Simple* has been upgraded from version 1.001003 to 1.001006.  
Various documentation fixes.
- *threads* has been upgraded from version 1.95 to 1.96.  
No changes to installed files other than version bumps.
- *Time::Piece* has been upgraded from version 1.27 to 1.29.  
When pretty printing negative *Time::Seconds*, the "minus" is no longer lost.
- *version* has been upgraded from version 0.9908 to 0.9909.  
Numerous changes. See the *Changes* file in the CPAN distribution for details.

## Documentation

## Changes to Existing Documentation

### perlfunc

- Calling `delete` or `exists` on array values is now described as "strongly discouraged" rather than "deprecated".

### perlpolicy

- The conditions for marking an experimental feature as non-experimental are now set out.

### perlrecharclass

- The documentation of Bracketed Character Classes has been expanded to cover the improvements in `qr/[ \N{named sequence} ]/` (see under *Selected Bug Fixes*).

### perlsyn

- An ambiguity in the documentation of the Ellipsis statement has been corrected. [*perl* #122661]

### perlxs

- Added a discussion of locale issues in XS code.

## 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 Warnings

- *Character in 'C' format overflow in pack*  
(W pack) You tried converting an infinity or not-a-number to an unsigned character, which makes no sense. Perl behaved as if you tried to pack 0xFF.
- *Character in 'c' format overflow in pack*  
(W pack) You tried converting an infinity or not-a-number to a signed character, which makes no sense. Perl behaved as if you tried to pack 0xFF.
- *Invalid number (%f) in chr*  
(W utf8) You passed an invalid number (like an infinity or not-a-number) to `chr`. Those are not valid character numbers, so it returned the Unicode replacement character (U+FFFD).

## Changes to Existing Diagnostics

- *Global symbol "%s" requires explicit package name*  
This message has had '(did you forget to declare "my %s"?)' appended to it, to make it more helpful to new Perl programmers. [*perl* #121638]
- *\N{} in character class restricted to one character in regex; marked by <-- HERE in m/%s/*  
This message has had 'character class' changed to 'inverted character class or as a range end-point is' to reflect improvements in `qr/[ \N{named sequence} ]/` (see under *Selected Bug Fixes*).
- *panic: frexp*  
This message has had ': %f' appended to it, to show what the offending floating point number is.

## Diagnostic Removals

- "Constant is not a FOO reference"  
Compile-time checking of constant dereferencing (e.g., `my_constant->()`) has been

removed, since it was not taking overloading into account. [[perl #69456](#)] [[perl #122607](#)]

- "Ambiguous use of -foo resolved as -&foo()"  
There is actually no ambiguity here, and this impedes the use of negated constants; e.g., -Inf.

## Configuration and Compilation

- For long doubles (to get more precision and range for floating point numbers) one can now use the GCC quadmath library which implements the quadruple precision floating point numbers in x86 and ia64 platforms. See *INSTALL* for details.

## Testing

- A new test script, *op/infnan.t*, has been added to test if Inf and NaN are working correctly. See *Infinity and NaN (not-a-number) handling improved*.
- A new test script, *re/rt122747.t*, has been added to test that the fix for *perl #122747* is working.

## Internal Changes

- `save_re_context` no longer does anything and has been moved to *mathoms.c*.
- `cv_name` is a new API function that can be passed a CV or GV. It returns an SV containing the name of the subroutine for use in diagnostics. [[perl #116735](#)] [[perl #120441](#)]
- `cv_set_call_checker_flags` is a new API function that works like `cv_set_call_checker`, except that it allows the caller to specify whether the call checker requires a full GV for reporting the subroutine's name, or whether it could be passed a CV instead. Whatever value is passed will be acceptable to `cv_name`. `cv_set_call_checker` guarantees there will be a GV, but it may have to create one on the fly, which is inefficient. [[perl #116735](#)]
- `CvGV` (which is not part of the API) is now a more complex macro, which may call a function and reify a GV. For those cases where it has been used as a boolean, `CvHASGV` has been added, which will return true for CVs that notionally have GVs, but without reifying the GV. `CvGV` also returns a GV now for lexical subs. [[perl #120441](#)]
- Added "*sync\_locale*" in *perlapi*. Changing the program's locale should be avoided by XS code. Nevertheless, certain non-Perl libraries called from XS, such as `Gtk` do so. When this happens, Perl needs to be told that the locale has changed. Use this function to do so, before returning to Perl.
- The defines and labels for the flags in the `op_private` field of OPs are now auto-generated from data in *regen/op\_private*. The noticeable effect of this is that some of the flag output of *Concise* might differ slightly, and the flag output of `perl -Dx` may differ considerably (they both use the same set of labels now). Also in debugging builds, there is a new assert in `op_free()` that checks that the op doesn't have any unrecognized flags set in `op_private`.

## Selected Bug Fixes

- Constant dereferencing now works correctly for typeglob constants. Previously the glob was stringified and its name looked up. Now the glob itself is used. [[perl #69456](#)]
- When parsing a funny character (`$ @ % &`) followed by braces, the parser no longer tries to guess whether it is a block or a hash constructor (causing a syntax error when it guesses the latter), since it can only be a block.
- `undef $reference` now frees the referent immediately, instead of hanging on to it until the next statement. [[perl #122556](#)]
- Various cases where the name of a sub is used (autoload, overloading, error messages) used to crash for lexical subs, but have been fixed.

- Bareword lookup now tries to avoid vivifying packages if it turns out the bareword is not going to be a subroutine name.
- Compilation of anonymous constants (e.g., `sub () { 3 }`) no longer deletes any subroutine named `__ANON__` in the current package. Not only was `*__ANON__ {CODE}` cleared, but there was a memory leak, too. This bug goes back to Perl 5.8.0.
- Stub declarations like `sub f;` and `sub f ();` no longer wipe out constants of the same name declared by `use constant`. This bug was introduced in Perl 5.10.0.
- Under some conditions a warning raised in compilation of regular expression patterns could be displayed multiple times. This is now fixed.
- `qr/[ \N{named sequence} ]/` now works properly in many instances. Some names known to `\N{...}` refer to a sequence of multiple characters, instead of the usual single character. Bracketed character classes generally only match single characters, but now special handling has been added so that they can match named sequences, but not if the class is inverted or the sequence is specified as the beginning or end of a range. In these cases, the only behavior change from before is a slight rewording of the fatal error message given when this class is part of a `?[...]` construct. When the `[...]` stands alone, the same non-fatal warning as before is raised, and only the first character in the sequence is used, again just as before.
- Tainted constants evaluated at compile time no longer cause unrelated statements to become tainted. [*perl #122669*]
- `open $$fh, ...`, which vivifies a handle with a name like "main::\_GEN\_0", was not giving the handle the right reference count, so a double free could happen.
- When deciding that a bareword was a method name, the parser would get confused if an "our" sub with the same name existed, and look up the method in the package of the "our" sub, instead of the package of the invocant.
- The parser no longer gets confused by `\U=` within a double-quoted string. It used to produce a syntax error, but now compiles it correctly. [*perl #80368*]
- It has always been the intention for the `-B` and `-T` file test operators to treat UTF-8 encoded files as text. (*perlfunc* has been updated to say this.) Previously, it was possible for some files to be considered UTF-8 that actually weren't valid UTF-8. This is now fixed. The operators now work on EBCDIC platforms as well.
- Under some conditions warning messages raised during regular expression pattern compilation were being output more than once. This has now been fixed.
- A regression has been fixed that was introduced in Perl 5.20.0 (fixed in Perl 5.20.1 as well as here) in which a UTF-8 encoded regular expression pattern that contains a single ASCII lowercase letter does not match its uppercase counterpart. [*perl #122655*]
- Constant folding could incorrectly suppress warnings if lexical warnings (`use warnings` or `no warnings`) were not in effect and `$^W` were false at compile time and true at run time.
- Loading UTF8 tables during a regular expression match could cause assertion failures under debugging builds if the previous match used the very same regular expression. [*perl #122747*]
- Thread cloning used to work incorrectly for lexical subs, possibly causing crashes or double frees on exit.
- Since Perl 5.14.0, deleting `$SomePackage:: {__ANON__}` and then undefining an anonymous subroutine could corrupt things internally, resulting in *Devel::Peek* crashing or *B.pm* giving nonsensical data. This has been fixed.



- `(caller $n)[3]` now reports names of lexical subs, instead of treating them as `"(unknown)"`.
- `sort subname LIST` now supports lexical subs for the comparison routine.
- Aliasing (e.g., via `*x = *y`) could confuse list assignments that mention the two names for the same variable on either side, causing wrong values to be assigned. [*perl #15667*]
- Long here-doc terminators could cause a bad read on short lines of input. This has been fixed. It is doubtful that any crash could have occurred. This bug goes back to when here-docs were introduced in Perl 3.000 twenty-five years ago.
- An optimization in `split` to treat `split/^/` like `split/^/m` had the unfortunate side-effect of also treating `split/\A/` like `split/^/m`, which it should not. This has been fixed. (Note, however, that `split/^x/` does not behave like `split/^x/m`, which is also considered to be a bug and will be fixed in a future version.) [*perl #122761*]
- The little-known `my Class $var` syntax (see *fields* and *attributes*) could get confused in the scope of `use utf8` if `Class` were a constant whose value contained Latin-1 characters.

## Acknowledgements

Perl 5.21.4 represents approximately 4 weeks of development since Perl 5.21.3 and contains approximately 29,000 lines of changes across 520 files from 30 authors.

Excluding auto-generated files, documentation and release tools, there were approximately 15,000 lines of changes to 390 .pm, .t, .c and .h files.

Perl continues to flourish into its third decade thanks to a vibrant community of users and developers. The following people are known to have contributed the improvements that became Perl 5.21.4:

Alberto Simões, Alexandre (Midnite) Jousset, Andy Dougherty, Anthony Heading, Brian Fraser, Chris 'BinGOs' Williams, Craig A. Berry, Daniel Dragan, David Mitchell, Doug Bell, Father Chrysostomos, George Greer, H.Merijn Brand, James E Keenan, Jarkko Hietaniemi, Jerry D. Hedden, Karen Etheridge, Karl Williamson, Olivier Mengué, Peter Martini, Reini Urban, Ricardo Signes, Steffen Müller, Steve Hay, Sullivan Beck, syber, Tadeusz Sońnierz, Tony Cook, Yves Orton, Årvar Arnfrid Bjarmason.

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

Many of the changes included in this version originated in the CPAN modules included in Perl's core. We're grateful to the entire CPAN community for helping Perl to flourish.

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

## Reporting Bugs

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

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

If the bug you are reporting has security implications, which make it inappropriate to send to a publicly archived mailing list, then please send it to `perl5-security-report@perl.org`. This points to a closed subscription unarchived mailing list, which includes all the core committers, who will be able to help assess the impact of issues, figure out a resolution, and help co-ordinate the release of patches to

mitigate or fix the problem across all platforms on which Perl is supported. Please only use this address for security issues in the Perl core, not for modules independently distributed on CPAN.

## SEE ALSO

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

The *INSTALL* file for how to build Perl.

The *README* file for general stuff.

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