

NAME

perl5217delta - what is new for perl v5.21.7

DESCRIPTION

This document describes differences between the 5.21.6 release and the 5.21.7 release.

If you are upgrading from an earlier release such as 5.21.5, first read *perl5216delta*, which describes differences between 5.21.5 and 5.21.6.

Core Enhancements

faster array and hash lookups

Array and hash lookups (especially nested ones) that use only constants or simple variables as keys, are now considerably faster. See *Internal Changes* for more details.

fileno now works on directory handles

When the relevant support is available in the operating system, the `fileno` builtin now works on directory handles, yielding the underlying file descriptor in the same way as for filehandles. On operating systems without such support, `fileno` on a directory handle continues to return the undefined value, as before, but also sets `$!` to indicate that the operation is not supported.

Currently, this uses either a `dd_fd` member in the OS `DIR` structure, or a `dirfd(3)` function as specified by POSIX.1-2008.

Incompatible Changes

& and \& prototypes accepts only subs

The `&` prototype character now accepts only anonymous subs (`sub { . . . }`) and things beginning with `\&`. Formerly it erroneously also allowed `undef` and references to array, hashes, and lists. [perl #4539] [perl #123062]

The `\&` prototype was allowing subroutine calls, whereas now it only allows subroutines. `&foo` is permitted. `&foo()` and `foo()` are not. [perl #77860]

use encoding is now lexical

The *encoding* pragma's effect is now limited to lexical scope. This pragma is deprecated, but in the meantime, it could adversely affect unrelated modules that are included in the same program.

List slices returning empty lists

List slices return an empty list now only if the original list was empty (or if there are no indices). Formerly, a list slice would return an empty list if all indices fell outside the original list. [perl #114498]

Deprecations

Setting \${^ENCODING} to anything but undef

This variable allows Perl scripts to be written in a non-ASCII, non-UTF-8 encoding. However, it affects all modules globally, leading to wrong answers and segmentation faults. New scripts should be written in UTF-8; old scripts should be converted to UTF-8, which is easily done with the *encoding* pragma.

Performance Enhancements

- Win32 Perl uses 8 KB less of per-process memory than before for every perl process of this version. This data is now memory mapped from disk and shared between perl processes from the same perl binary.
- If method and class names are known at compile time, hashes are precomputed to speed up run-time method lookup. Also, compound method names like `SUPER::new` are parsed at compile time, to save having to parse them at run time.

Modules and Pragmata

Updated Modules and Pragmata

- The PathTools module collection has been upgraded from 3.52 to 3.53.
Don't turn leading `//` into `/` on Cygwin. [perl #122635]
- *Archive::Tar* has been upgraded from version 2.02 to 2.04.
- *anybase* has been upgraded from version 0.08 to 0.09.
- *B* has been upgraded from version 1.53 to 1.54.
Nulled COPs are now of class `B::COP`, rather than `B::OP`.
`B::REGEXP` objects now provide a `qr_anoncv` method for accessing the implicit CV associated with `qr//` things containing code blocks, and a `compflags` method that returns the pertinent flags originating from the `qr//blahblah` op.
`B::PMOP` now provides a `pmregex` method returning a `B::REGEXP` object.
Two new classes, `B::PADNAME` and `B::PADNAMELIST`, have been introduced.
- *B::Concise* has been upgraded from version 0.995 to 0.996.
- *B::Deparse* has been upgraded from version 1.30 to 1.31.
BEGIN blocks at the end of the enclosing scope are now deparsed in the right place. [perl #77452]
BEGIN blocks were sometimes deparsed as `__ANON__`, but are now always called BEGIN.
Lexical subroutines are now fully deparsed. [perl #116553]
Deparsing of `$lexical =~ //` was accidentally broken in 1.30 (perl 5.21.6), omitting the `$lexical =~`, but has now been fixed.
Anything `=~ y///r` with `/r` no longer omits the left-hand operand.
The op trees that make up regexp code blocks are now deparsed for real. Formerly, the original string that made up the regular expression was used. That caused problems with `qr/(?{<heredoc})/` and multiline code blocks, which were deparsed incorrectly. [perl #123217] [perl #115256]
`$;` at the end of a statement no longer loses its semicolon. [perl #123357]
Some cases of subroutine declarations stored in the stash in shorthand form were being omitted.
Non-ASCII characters are now consistently escaped in strings, instead of some of the time. (There are still outstanding problems with regular expressions and identifiers that have not been fixed.)
When prototype sub calls are deparsed with `&` (e.g., under the **-P** option), `scalar` is now added where appropriate, to force the scalar context implied by the prototype.
`require(foo()), do(foo()), goto(foo())` and similar constructs with loop controls are now deparsed correctly. The outer parentheses are not optional.
Whitespace is no longer escaped in regular expressions, because it was getting erroneously escaped within `(?x: ...)` sections.
`sub foo { foo() }` is now deparsed with those mandatory parentheses.
`/@array/` is now deparsed as a regular expression, and not just `@array`.
`/@{-}/`, `/@{+}/` and `/${1}` are now deparsed with the braces, which are mandatory in these cases.
In deparsing feature bundles, `B::Deparse` was emitting `no feature; first` instead of `no feature 'all';`. This has been fixed.
`chdir FH` is now deparsed without quotation marks.

`\my @a` is now deparsed without parentheses. (Parentheses would flatten the array.)

`system` and `exec` followed by a block are now deparsed correctly. Formerly there was an erroneous `do` before the block.

use constant `QR => qr/.../flags` followed by `" " =~ QR` is no longer without the flags.

- *chardnames* has been upgraded from version 1.41 to 1.43.
- *Compress::Raw::Bzip2* has been upgraded from version 2.066 to 2.067.
- *Compress::Raw::Zlib* has been upgraded from version 2.066 to 2.067.
- *Data::Dumper* has been upgraded from version 2.154 to 2.155.
- *DB_File* has been upgraded from version 1.831 to 1.834.
- *Devel::Peek* has been upgraded from version 1.19 to 1.20.
- *Devel::PPPort* has been upgraded from version 3.24 to 3.25.
- *DynaLoader* has been upgraded from version 1.28 to 1.29.
- *Encode* has been upgraded from version 2.64 to 2.67.
- *Errno* has been upgraded from version 1.21 to 1.22.
- *ExtUtils::CBuilder* has been upgraded from version 0.280220 to 0.280221.
- *ExtUtils::MakeMaker* has been upgraded from version 7.02 to 7.04.
- *ExtUtils::ParseXS* has been upgraded from version 3.26 to 3.27.

Only declare `file` unused if we actually define it.

Improve generated `RETVAL` code generation to avoid repeated references to `ST(0)`. [perl #123278]

Broaden and document the `/OBJ$/` to `/REF$/` typemap optimization for the `DESTROY` method. [perl #123418]

- *feature* has been upgraded from version 1.38 to 1.39.
- *File::Spec* has been upgraded from version 3.51 to 3.54.
- *Filter::Util::Call* has been upgraded from version 1.49 to 1.51.
- *HTTP::Tiny* has been upgraded from version 0.051 to 0.053.
- *IO* has been upgraded from version 1.34 to 1.35.
- The *IO::Compress* module collection has been upgraded from version 2.066 to 2.067.
- *IO::Socket::IP* has been upgraded from version 0.32 to 0.34.
- *Locale::Codes* has been upgraded from version 3.32 to 3.33.
- *Locale::Maketext* has been upgraded from version 1.25 to 1.26.
- *Module::CoreList* has been upgraded from version 5.20141120 to 5.20141220.
- *mro* has been upgraded from version 1.16 to 1.17.
- *NDBM_File* has been upgraded from version 1.13 to 1.14.
- *Opcodes* has been upgraded from version 1.29 to 1.30.

- *PerlIO::scalar* has been upgraded from version 0.20 to 0.21.
Reading from a position well past the end of the scalar now correctly returns end of file. [perl #123443]
Seeking to a negative position still fails, but no longer leaves the file position set to a negation location.
`eof()` on a `PerlIO::scalar` handle now properly returns true when the file position is past the 2GB mark on 32-bit systems.
- *Pod::Escapes* has been upgraded from version 1.06 to 1.07.
- *POSIX* has been upgraded from version 1.46 to 1.48.
- *SDBM_File* has been upgraded from version 1.11 to 1.12.
- *Storable* has been upgraded from version 2.51 to 2.52.
- *strict* has been upgraded from version 1.08 to 1.09.
- *Sys::Hostname* has been upgraded from version 1.19 to 1.20.
- *Test::Simple* has been upgraded from version 1.301001_075 to 1.301001_090.
- *threads* has been upgraded from version 1.96 to 1.96_001.
- *Unicode::Collate* has been upgraded from version 1.07 to 1.09.
- *XSLoader* has been upgraded from version 0.19 to 0.20
Don't test twice for bootstrap file.

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

- "Variable %s will not stay shared" has been changed to say "Subroutine" when it is actually a lexical sub that will not stay shared.

Platform Support

Platform-Specific Notes

Win32

- Intel C builds are now always built with C99 mode on.

Internal Changes

- The `PADNAME` and `PADNAMELIST` types are now separate types, and no longer simply aliases for `SV` and `AV`. [perl #123223]
- Pad names are now always UTF8. The `PadnameUTF8` macro always returns true. Previously, this was effectively the case already, but any support for two different internal representations of pad names has now been removed.
- The `OP_SIBLING` and `OP_HAS_SIBLING` macros added in an earlier 5.21.x release have been renamed `OpSIBLING` and `OpHAS_SIBLING`, following the existing convention.
- A new op class, `UNOP_AUX`, has been added. This is a subclass of `UNOP` with an `op_aux` field added, which points to an array of unions of `UV`, `SV*` etc. It is intended for where an op needs to store more data than a simple `op_sv` or whatever. Currently the only op of this type is `OP_MULTIDEREf` (see below).
- A new op has been added, `OP_MULTIDEREf`, which performs one or more nested array and

hash lookups where the key is a constant or simple variable. For example the expression `$a[0][$k][$i]`, which previously involved `ten rv2Xv`, `Xelem`, `gvsv` and `const` ops is now performed by a single `multideref` op. It can also handle `local`, `exists` and `delete`. A non-simple index expression, such as `[$i+1]` is still done using `aelem/helem`, and single-level array lookup with a small constant index is still done using `aelemfast`.

Selected Bug Fixes

- On Win32, restoring in a child pseudo-process a variable that was `local()`ed in a parent pseudo-process before the `fork` happened caused memory corruption and a crash in the child pseudo-process (and therefore OS process). [perl #40565]
- Calling `write` on a format with a `^**` field could produce a panic in `sv_chop()` if there were insufficient arguments or if the variable used to fill the field was empty. [perl #123245]
- Non-ASCII lexical sub names (use in error messages) no longer have extra junk on the end.
- The `\@` subroutine prototype no longer flattens parenthesized arrays (taking a reference to each element), but takes a reference to the array itself. [perl #47363]
- A block containing nothing except a C-style `for` loop could corrupt the stack, causing lists outside the block to lose elements or have elements overwritten. This could happen with `map { for(...){...} } ...` and with lists containing `do { for(...){...} }`. [perl #123286]
- `scalar()` now propagates lvalue context, so that `for(scalar($#foo)) { ... }` can modify `$#foo` through `$_`.
- `qr/@array(?{block})/` no longer dies with "Bizarre copy of ARRAY". [#123344]
- `eval '$variable'` in nested named subroutines would sometimes look up a global variable even with a lexical variable in scope.
- In perl 5.20.0, `sort CORE::fake` where 'fake' is anything other than a keyword started chopping of the last 6 characters and treating the result as a sort sub name. The previous behaviour of treating "CORE::fake" as a sort sub name has been restored. [perl #123410]
- Outside of `use utf8`, a single-character Latin-1 lexical variable is disallowed. The error message for it, "Can't use global \$foo...", was giving garbage instead of the variable name.
- `readline` on a nonexistent handle was causing `$_{^LAST_FH}` to produce a reference to an undefined scalar (or fail an assertion). Now `$_{^LAST_FH}` ends up undefined.
- `(...)x...` in void context now applies scalar context to the left-hand argument, instead of the context the current sub was called in. [perl #123020]

Acknowledgements

Perl 5.21.7 represents approximately 4 weeks of development since Perl 5.21.6 and contains approximately 97,000 lines of changes across 640 files from 30 authors.

Excluding auto-generated files, documentation and release tools, there were approximately 26,000 lines of changes to 400 .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 5.21.7:

Aaron Crane, Aristotle Pagaltzis, Chad Granum, Chris 'BinGOs' Williams, Craig A. Berry, Dagfinn Ilmari Mannsåker, Daniel Dragan, David Mitchell, Eric Herman, Father Chrysostomos, H.Merijn Brand, Hugo van der Sanden, James Raspass, Jarkko Hietaniemi, Jim Cromie, Karen Etheridge, Karl Williamson, kmx, Lukas Mai, Matthew Horsfall, Max Maischein, Petr Páľsa™, Ricardo Signes, Shlomi Fish, Steffen Müller, Steve Hay, Sullivan Beck, syber, Todd Rinaldo, Tony Cook.

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.