

NAME

perl5135delta - what is new for perl v5.13.5

DESCRIPTION

This document describes differences between the 5.13.4 release and the 5.13.5 release.

If you are upgrading from an earlier release such as 5.13.3, first read *perl5134delta*, which describes differences between 5.13.3 and 5.13.4.

Core Enhancements

Adjacent pairs of nextstate opcodes are now optimized away

Previously, in code such as

```
use constant DEBUG => 0;

sub GAK {
    warn if DEBUG;
    print "stuff\n";
}
```

the ops for `warn if DEBUG;` would be folded to a null op (`ex-const`), but the `nextstate` op would remain, resulting in a runtime op dispatch of `nextstate, nextstate, ...`

The execution of a sequence of `nextstate` ops is indistinguishable from just the last `nextstate` op so the peephole optimizer now eliminates the first of a pair of `nextstate` ops, except where the first carries a label, since labels must not be eliminated by the optimizer and label usage isn't conclusively known at compile time.

API function to parse statements

The `parse_fullstmt` function has been added to allow parsing of a single complete Perl statement. See *perlapi* for details.

API functions for accessing the runtime hinthash

A new C API for introspecting the hinthash `%^H` at runtime has been added. See `cop_hints_2hv`, `cop_hints_fetchpvn`, `cop_hints_fetchpvs`, `cop_hints_fetchsv`, and `hv_copy_hints_hv` in *perlapi* for details.

C interface to caller()

The `caller_cx` function has been added as an XSUB-writer's equivalent of `caller()`. See *perlapi* for details.

Incompatible Changes

Magic variables outside the main package

In previous versions of Perl, magic variables like `$!`, `%SIG`, etc. would 'leak' into other packages. So `%foo::SIG` could be used to access signals, `${"foo::!"}` (with strict mode off) to access C's `errno`, etc.

This was a bug, or an 'unintentional' feature, which caused various ill effects, such as signal handlers being wiped when modules were loaded, etc.

This has been fixed (or the feature has been removed, depending on how you see it).

Smart-matching against array slices

Previously, the following code resulted in a successful match:

```
my @a = qw(a y0 z);
```

```
my @b = qw(a x0 z);
@a[0 .. $#b] ~~ @b;
```

This odd behaviour has now been fixed [*perl #77468*].

C API changes

The first argument of the C API function `Perl_fetch_cop_label` has changed from `struct refcounted he *` to `COP *`, to better insulate the user from implementation details.

This API function was marked as "may change", and likely isn't in use outside the core. (Neither an unpacked CPAN, nor Google's codesearch, finds any other references to it.)

Deprecations

Use of `qw(...)` as parentheses

Historically the parser fooled itself into thinking that `qw(. . .)` literals were always enclosed in parentheses, and as a result you could sometimes omit parentheses around them:

```
for $x qw(a b c) { ... }
```

The parser no longer lies to itself in this way. Wrap the list literal in parentheses, like:

```
for $x (qw(a b c)) { ... }
```

Performance Enhancements

- Scalars containing regular expressions now only allocate the part of the `SV` body they actually use, saving some space.
- Compiling regular expressions has been made faster for the case where upgrading the regex to utf8 is necessary but that isn't known when the compilation begins.

Modules and Pragmata

Updated Modules and Pragmata

`bignum`

Upgraded from version 0.23 to 0.25.

`blib`

Upgraded from version 1.05 to 1.06.

`open`

Upgraded from version 1.07 to 1.08.

`threads-shared`

Upgraded from version 1.33_02 to 1.33_03.

`warnings` and `warnings::register`

Upgraded from version 1.10 to 1.11 and from version 1.01 to 1.02 respectively.

It is now possible to register warning categories other than the names of packages using `warnings::register`. See *perllexwarn* for more information.

`B::Debug`

Upgraded from version 1.12 to 1.16.

`CPANPLUS::Dist::Build`

Upgraded from version 0.46 to 0.48.

Data::Dumper

Upgraded from version 2.126 to 2.128.

This fixes a crash when using custom sort functions that might cause the stack to change.

Encode

Upgraded from version 2.39 to 2.40.

Errno

Upgraded from version 1.12 to 1.13.

On some platforms with unusual header files, like Win32/gcc using mingw64 headers, some constants which weren't actually error numbers have been exposed by `Errno`. This has been fixed *[perl #77416]*.

ExtUtils::MakeMaker

Upgraded from version 6.5601 to 6.57_05.

Filter::Simple

Upgraded from version 0.84 to 0.85.

Hash::Util

Upgraded from version 0.08 to 0.09.

Math::BigInt

Upgraded from version 1.89_01 to 1.95.

This fixes, among other things, incorrect results when computing binomial coefficients *[perl #77640]*.

Math::BigInt::FastCalc

Upgraded from version 0.19 to 0.22.

Math::BigRat

Upgraded from version 0.24 to 0.26.

Module::CoreList

Upgraded from version 2.37 to 2.38.

PerlIO::scalar

Upgraded from version 0.08 to 0.09.

POSIX

Upgraded from version 1.19 to 1.20.

It now includes constants for POSIX signal constants.

Safe

Upgraded from version 2.27 to 2.28.

This fixes a possible infinite loop when looking for coderefs.

Test::Simple

Upgraded from version 0.96 to 0.97_01.

Tie::Hash

Upgraded from version 1.03 to 1.04.

Calling `Tie::Hash->TIEHASH()` used to loop forever. Now it croaks.

Unicode::Collate

Upgraded from version 0.56 to 0.59.

XSLoader

Upgraded from version 0.10 to 0.11.

Documentation

Changes to Existing Documentation

perlapi

- Many of the optree construction functions are now documented.

perlbook

- Expanded to cover many more popular books.

perlfaq

- *perlfaq*, *perlfaq2*, *perlfaq4*, *perlfaq5*, *perlfaq6*, *perlfaq8*, and *perlfaq9* have seen various updates and modernizations.

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

- Parsing code internal error (%s)
New fatal error produced when parsing code supplied by an extension violated the parser's API in a detectable way.
- Use of `qw(...)` as parentheses is deprecated
See *Use of qw(...) as parentheses* for details.

Changes to Existing Diagnostics

- `warn` and `die` now produce 'Wide character' warnings when fed a character outside the byte range if `STDERR` is a byte-sized handle.

Utility Changes

h2ph

- The use of a deprecated `goto` construct has been removed [*perl #74404*].

Testing

- The new *t/lib/universal.t* script tests the `Internal::*` functions and other things in *universal.c*.
- A rare race condition in *t/op/while_readdir.t* has been fixed, stopping it from failing randomly when running tests in parallel.
- The new *t/op/leaky-magic.t* script tests that magic applied to variables in the main packages does not affect other packages.

Platform Support

Platform-Specific Notes

VMS

- Make `PerlIOUnix_open` honour default permissions on VMS.
When `perlio` became the default and `unixio` became the default bottom layer, the most common path for creating files from Perl became `PerlIOUnix_open`, which has always explicitly used `0666` as the permission mask.

To avoid this, 0777 is now passed as the permissions to `open()`. In the VMS CRTL, 0777 has a special meaning over and above intersecting with the current umask; specifically, it allows Unix syscalls to preserve native default permissions.

Internal Changes

- `CALL_FPTR` and `CPERLscope` have been deprecated.
Those are left from an old implementation of `MULTIPLICITY` using C++ objects, which was removed in Perl 5.8. Nowadays these macros do exactly nothing, so they shouldn't be used anymore.
For compatibility, they are still defined for external XS code. Only extensions defining `PERL_CORE` must be updated now.
- `lex_stuff_pvs()` has been added as a convenience macro wrapping `lex_stuff_pvn()` for literal strings.
- The recursive part of the peephole optimizer is now hookable.
In addition to `PL_peepp`, for hooking into the toplevel peephole optimizer, a `PL_rpeepp` is now available to hook into the optimizer recursing into side-chains of the optree.

Selected Bug Fixes

- A regression introduced in Perl 5.12.0, making `my $x = 3; $x = length(undef)` result in `$x` set to 3 has been fixed. `$x` will now be `undef`.
- A fatal error in regular expressions when processing UTF-8 data has been fixed [*perl #75680*].
- An erroneous regular expression engine optimization that caused regex verbs like `*COMMIT` to sometimes be ignored has been removed.
- The Perl debugger now also works in taint mode [*perl #76872*].
- Several memory leaks in cloning and freeing threaded Perl interpreters have been fixed [*perl #77352*].
- A possible string corruption when doing regular expression matches on overloaded objects has been fixed [*perl #77084*].
- Magic applied to variables in the main package no longer affects other packages. See *Magic variables outside the main package* above [*perl #76138*].
- Opening a glob reference via `open $fh, ">", *glob` will no longer cause the glob to be corrupted when the filehandle is printed to. This would cause perl to crash whenever the glob's contents were accessed [*perl #77492*].
- The postincrement and postdecrement operators, `++` and `--`, used to cause leaks when being used on references. This has now been fixed.
- A bug when replacing the glob of a loop variable within the loop has been fixed [*perl #21469*]. This means the following code will no longer crash:

```
for $x (...) {  
    *x = *y;  
}
```
- Perl would segfault if the undocumented `Internals` functions that used reference prototypes were called with the `&foo()` syntax, e.g. `&Internals::SvREADONLY(undef)` [*perl #77776*].
These functions now call `SvROK` on their arguments before dereferencing them with `SvRV`, and we test for this case in *t/lib/universal.t*.

- When assigning a list with duplicated keys to a hash, the assignment used to return garbage and/or freed values:

```
@a = %h = (list with some duplicate keys);
```

This has now been fixed [perl #31865].

- An earlier release of the 5.13 series of Perl changed the semantics of opening a reference to a copy of a glob:

```
my $var = *STDOUT;  
open my $fh, '>', \ $var;
```

This was a mistake, and the previous behaviour from Perl 5.10 and 5.12, which is to treat \ \$var as a scalar reference, has now been restored.

- The regular expression bracketed character class [\8\9] was effectively the same as [89\000], incorrectly matching a NULL character. It also gave incorrect warnings that the 8 and 9 were ignored. Now [\8\9] is the same as [89] and gives legitimate warnings that \8 and \9 are unrecognized escape sequences, passed-through.
- warn() and die() now respect utf8-encoded scalars [perl #45549].

Known Problems

- The upgrade to Encode-2.40 has caused some tests in the libwww-perl distribution on CPAN to fail. (Specifically, *base/message-charset.t* tests 33-36 in version 5.836 of that distribution now fail.)
- The upgrade to ExtUtils-MakeMaker-6.57_05 has caused some tests in the Module-Install distribution on CPAN to fail. (Specifically, *02_mymeta.t* tests 5 and 21, *18_all_from.t* tests 6 and 15, *19_authors.t* tests 5, 13, 21 and 29, and *20_authors_with_special_characters.t* tests 6, 15 and 23 in version 1.00 of that distribution now fail.)

Acknowledgements

Perl 5.13.5 represents approximately one month of development since Perl 5.13.4 and contains 74558 lines of changes across 549 files from 45 authors and committers:

Abigail, Alexander Alekseev, Aristotle Pagaltzis, Ben Morrow, Bram, brian d foy, Chas. Owens, Chris 'BinGOs' Williams, Craig A. Berry, Curtis Jewell, Dagfinn Ilmarinen, David Golden, David Leadbeater, David Mitchell, Eric Brine, Father Chrysostomos, Florian Ragwitz, Gisle Aas, Jan Dubois, Jerry D. Hedden, Jesse Vincent, Jim Cromie, Jirka Hruška, Karl Williamson, Michael G. Schwern, Nicholas Clark, Paul Johnson, Philippe Bruhat (Book), Piotr Fusik, Rafael Garcia-Suarez, Rainer Tammer, Reini Urban, Ricardo Signes, Rob Hoelz, Robin Barker, Steffen Mueller, Steve Hay, Steve Peters, Todd Rinaldo, Tony Cook, Vincent Pit, Yves Orton, Zefram, Zsolt Ambrus, Átvar Arnfríð Bjarmason.

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.

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 <http://rt.perl.org/perlbug/>. 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 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.