

NAME

perl51311delta - what is new for perl v5.13.11

DESCRIPTION

This document describes differences between the 5.13.10 release and the 5.13.11 release.

If you are upgrading from an earlier release such as 5.13.9, first read *perl5139delta*, which describes differences between 5.13.9 and 5.13.10.

Security

User-defined regular expression properties

Perl no longer allows a tainted regular expression to invoke a user-defined property via `\p{...}` syntax. It simply dies instead [perl #82616].

Incompatible Changes

local(\$_) will strip all magic from \$_

`local()` on scalar variables will give them a new value, but keep all their magic intact. This has proven to be problematic for the default scalar variable `$_`, where *perlsub* recommends that any subroutine that assigns to `$_` should localize it first. This would throw an exception if `$_` is aliased to a read-only variable, and could have various unintentional side-effects in general.

Therefore, as an exception to the general rule, `local($_)` will not only assign a new value to `$_`, but also remove all existing magic from it as well.

Passing references to warn()

An earlier Perl 5.13.x release changed `warn($ref)` to leave the reference unchanged, allowing `$SIG{__WARN__}` handlers to access the original reference. But this stopped warnings that were references from having the file and line number appended even when there was no `$SIG{__WARN__}` handler in place.

Now `warn` checks for the presence of such a handler and, if there is none, proceeds to stringify the reference and append the file and line number. This allows simple uses of `warn` for debugging to continue to work as they did before.

fork() emulation will not wait for signalled children

On Windows parent processes would not terminate until all forked children had terminated first. However, `kill('KILL', ...)` is inherently unstable on pseudo-processes, and `kill('TERM', ...)` might not get delivered if the child is blocked in a system call.

To avoid the deadlock and still provide a safe mechanism to terminate the hosting process, Perl will now no longer wait for children that have been sent a `SIGTERM` signal. It is up to the parent process to `waitpid()` for these children if child clean-up processing must be allowed to finish. However, it is also the responsibility of the parent then to avoid the deadlock by making sure the child process can't be blocked on I/O either.

See *perlfork* for more information about the `fork()` emulation on Windows.

Perl source code is read in text mode on Windows

Perl scripts used to be read in binary mode on Windows for the benefit of the `ByteLoader` module (which is no longer part of core Perl). This had the side effect of breaking various operations on the `DATA` filehandle, including `seek()/tell()`, and even simply reading from `DATA` after file handles have been flushed by a call to `system()`, `backticks`, `fork()` etc.

The default build options for Windows have been changed to read Perl source code on Windows in text mode now. Hopefully `ByteLoader` will be updated on CPAN to automatically handle this situation.

Performance Enhancements

- An earlier optimisation to speed up `my @array = ...` and `my %hash = ...` assignments caused a bug and was disabled in Perl 5.12.0.

Now we have found another way to speed up these assignments [perl #82110].

Modules and Pragmata

Updated Modules and Pragmata

- `attributes` has been upgraded from version 0.13 to 0.14.
- `base` has been upgraded from version 2.15 to 2.16.
- `CPAN` has been upgraded from version 1.94_65 to 1.9600.
- `CPANPLUS` has been upgraded from version 0.9101 to 0.9103
- `CPANPLUS::Dist::Build` has been upgraded from version 0.52 to 0.54
- `Cwd` has been downgraded from version 3.37 to 3.36.
An optimisation that recent core changes have rendered unnecessary has been reverted.
- `Devel::DProf` has been upgraded from version 20110225.01 to 20110228.00.
- `Digest::SHA` has been upgraded from version 5.50 to 5.61
New SHA-512/224 and SHA-512/256 transforms ref. NIST Draft FIPS 180-4 (February 2011)
- `ExtUtils::Command` has been upgraded from version 1.16 to 1.17.
- `File::Copy` has been downgraded from version 2.22 to 2.21.
An optimisation that recent core changes have rendered unnecessary has been reverted.
- `File::Glob` has been upgraded from version 1.11 to 1.12.
- `GDBM_File` has been upgraded from version 1.13 to 1.14.
- `Hash::Util` has been upgraded from version 0.10 to 0.11.
- `Hash::Util::FieldHash` has been upgraded from version 1.08 to 1.09.
- `HTTP::Tiny` has been upgraded from version 0.010 to 0.011.
- `I18N::Langinfo` has been upgraded from version 0.07 to 0.08.
- `IO` has been upgraded from version 1.25_03 to 1.25_04.
- `JSON::PP` has been upgraded from version 2.27103 to 2.27105
- `Locale::Codes` has been upgraded from version 3.15 to 3.16
- `Math::BigInt` has been upgraded from version 1.992 to 1.994
- `Math::BigInt::FastCalc` has been upgraded from version 0.24_02 to 0.28
- `Module::Build` has been upgraded from version 0.37_05 to 0.3800
- `Module::CoreList` has been upgraded from version 2.45 to 2.46.
- `mro` has been upgraded from version 1.06 to 1.07.
- `NDBM_File` has been upgraded from version 1.11 to 1.12.
- `parent` has been upgraded from version 0.224 to 0.225
- `Pod::Simple` has been upgraded from version 3.15 to 3.16

- `Storable` has been upgraded from version 2.26 to 2.27.
- `Sys::Hostname` has been upgraded from version 1.15 to 1.16.
- `Test::Harness` has been upgraded from version 3.22 to 3.23
- `Test::Simple` has been upgraded from version 0.97_01 to 0.98
- `Tie::Hash::NamedCapture` has been upgraded from version 0.07 to 0.08.
Some of the Perl code has been converted to XS for efficiency's sake.
- `Tie::RefHash` has been upgraded from version 1.38 to 1.39.
- `Unicode::Collate` has been upgraded from version 0.72 to 0.73
DUCET has been updated for Unicode 6.0.0 as `Collate/allkeys.txt` and the default `UCA_Version` is 22.
- `Unicode::UCD` has been upgraded from version 0.31 to 0.32. This includes a number of bug fixes:
 - `charinfo()`
 - It is now updated to Unicode Version 6 with Corrigendum #8, except, as with Perl 5.14, the code point at U+1F514 has no name.
 - The Hangul syllable code points have the correct names, and their decompositions are always output without requiring *Lingua::KO::Hangul::Util* to be installed.
 - The CJK (Chinese-Japanese-Korean) code points U+2A700 - U+2B734 and U+2B740 - 2B81D are now properly handled.
 - The numeric values are now output for those CJK code points that have them.
 - The names that are output for code points with multiple aliases are now the corrected ones.
 - `charscript()`
This now correctly returns "Unknown" instead of `undef` for the script of a code point that hasn't been assigned another one.
 - `charblock()`
This now correctly returns "No_Block" instead of `undef` for the block of a code point that hasn't been assigned to another one.
- `XS::TypeMap` has been upgraded from version 0.04 to 0.05.

Documentation

Changes to Existing Documentation

`perlfunc`

- Clarified the order in which to check `$@` and `$!` after `do FILE`. (RT #80626)

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

- Regex modifier `"/%c"` may not appear twice
(F syntax) The regular expression pattern had one of the mutually exclusive modifiers

repeated. Remove all but one of the occurrences.

- Regexp modifiers `"/%c"` and `"/%C"` are mutually exclusive
(F syntax) The regular expression pattern had more than one of the mutually exclusive modifiers. Retain only the modifier that is supposed to be there.
- Insecure user-defined property `%s`
(F) Perl detected tainted data when trying to compile a regular expression that contains a call to a user-defined character property function, i.e. `\p{IsFoo}` or `\p{InFoo}`. See *"User-Defined Character Properties" in perlunicode and perlsec*.

Testing

Many of the tests have been refactored to use testing libraries more consistently. In some cases test files were created or deleted:

- The tests for `split /\s/` and Unicode have been moved from `t/op/split.t` to the new `t/op/split_unicode.t`.
- `t/re/re.t` has been moved to `ext/re/t/re_funcs_u.t`.
- The tests for [perl #72922] have been moved from `t/re/qr.t` to the new `t/re/qr-72922.t`.
- `t/re/reg_unsafe.t` has been deleted and its only test moved to `t/re/pat_advanced.t`.

Selected Bug Fixes

- A fix for a bug in `length(undef)` in 5.13.4 introduced a regression that meant `print length undef` did not warn when warnings were enabled. It now correctly warns [perl #85508].
- The `(?|...)` regular expression construct no longer crashes if the final branch has more sets of capturing parentheses than any other branch. This was fixed in Perl 5.10.1 for the case of a single branch, but that fix did not take multiple branches into account [perl #84746].
- Accessing an element of a package array with a hard-coded number (as opposed to an arbitrary expression) would crash if the array did not exist. Usually the array would be autovivified during compilation, but typeglob manipulation could remove it, as in these two cases which used to crash:

```
*d = *a; print $d[0];
undef *d; print $d[0];
```
- `#line` directives in string evals were not properly updating the arrays of lines of code (`@{"_<..."}`) that the debugger (or any debugging or profiling module) uses. In threaded builds, they were not being updated at all. In non-threaded builds, the line number was ignored, so any change to the existing line number would cause the lines to be misnumbered [perl #79442].
- `$AUTOLOAD` used to remain tainted forever if it ever became tainted. Now it is correctly untainted if an autoloader method is called and the method name was not tainted.
- A bug has been fixed in the implementation of `{...}` quantifiers in regular expressions that prevented the code block in `/((\w+)(?{ print $2 })){2}/` from seeing the `$2` sometimes [perl #84294].
- `sprintf` now dies when passed a tainted scalar for the format. It did already die for arbitrary expressions, but not for simple scalars [perl #82250].
- `DESTROY` methods of objects implementing ties are no longer able to crash by accessing the tied variable through a weak reference [perl #86328].

- On Windows, calling `kill(9, $child)` on a pseudo-process created by the `fork()` emulation is inherently unstable. It can also be responsible for overriding the parent process exit code with a value of '9' if the parent terminates right after killing the child. This condition will now happen a lot less often than before.

See also *fork() emulation will not wait for signalled children* for a better way to terminate child processes that avoids deadlocks altogether.

- Ensure that the `exists &Errno::EFOO` idiom continues to work as documented.

A change post-5.12 caused the documented idiom not to work if `Errno` was loaded after the `exists` code had been compiled, as the compiler implicitly creates typeglobs in the `Errno` symbol table when it builds the optree for the `exists` code.

Acknowledgements

Perl 5.13.11 represents approximately one month of development since Perl 5.13.10 and contains approximately 80,000 lines of changes across 549 files from 31 authors and committers:

Alastair Douglas, Arvan, Boris Ratner, brian d foy, Chris 'BinGOs' Williams, Craig A. Berry, David Golden, David Leadbeater, David Mitchell, Father Chrysostomos, Florian Ragwitz, Jan Dubois, Karl Williamson, Kevin Ryde, Leon Brocard, Leon Timmermans, Michael Stevens, Michael Witten, Moritz Lenz, Nicholas Clark, Paul Johnson, Peter John Acklam, Reini Urban, Robin Barker, Steve Hay, Sullivan Beck, Tony Cook, Vadim Konovalov, Yves Orton, Zefram and `Ã†var ArnfjÃ¶rÃ° Bjarmason`

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.