## NAME

perl51310delta - what is new for perl v5.13.10

## DESCRIPTION

This document describes differences between the 5.13.9 release and the 5.13.10 release.

If you are upgrading from an earlier release such as 5.13.8, first read *perl5139delta*, which describes differences between 5.13.8 and 5.13.9.

## Core Enhancements

### The new regular expression modifiers available in suffix form

Various releases of the 5.13.x series have added new regular expression modifiers, `/a`, `/d`, `/l`, and `/u`. They were only available in infix form (e.g., `(?a:...)`) until this release; now they are usable in suffix form. This change was made too late to change all the affected documentation, so there are a number of places that erroneously say these must be used in infix form.

However, there is an ambiguity with the construct, `s/foo/bar/le...`. Due to backward compatibility constraints, in Perl 5.14 only, it will be resolved as `s/foo/bar/ le...`, that is, as meaning to take the result of the substitution, and see if it is stringwise less-than-or-equal-to what follows. In Perl 5.16 and later, it will instead be resolved as meaning to do the pattern match using the rules of the current locale, and evaluate the rhs as an expression when doing the substitution. In 5.14, if you want the latter interpretation, you can write "el" instead.

### Add \p{Titlecase} as a synonym for \p{Title}

This synonym is added for symmetry with the Unicode property names `\p{Uppercase}` and `\p{Lowercase}`.

### New regular expression modifier option /aa

Doubling the `/a` regular expression modifier increases its effect, so that in case-insensitive matching, no ASCII character will match a non-ASCII character. For example, normally,

```
    'k' =~ /\N{KELVIN SIGN}/i
```

will match; it won't under `/aa`.

### New warnings categories for problematic (non-)Unicode code points.

Three new warnings subcategories of <utf8> have been added. These allow you to turn off warnings for their covered events, while allowing the other UTF-8 warnings to remain on. The three categories are: `surrogate` when UTF-16 surrogates are encountered; `nonchar` when Unicode non-character code points are encountered; and `non_unicode` when code points that are above the legal Unicode maximum of 0x10FFFF are encountered.

## Incompatible Changes

### Most \p{} properties are now immune from case-insensitive matching

For most Unicode properties, it doesn't make sense to have them match differently under `/i` case-insensitive matching than not. And doing so leads to unexpected results and potential security holes. For example

```
 m/\p{ASCII_Hex_Digit}+/i
```

could previously match non-ASCII characters because of the Unicode matching rules. There were a number of bugs in this feature until an earlier release in the 5.13 series. Now this release reverts, and removes the feature completely except for the few properties where people have come to expect it, namely the ones where casing is an integral part of their functionality, such as `m/\p{Uppercase}/i` and `m/\p{Lowercase}/i`, both of which match the exact same code points, namely those matched by `m/\p{Cased}/i`. Details are in *"Unicode Properties" in perlrecharclass*.

User-defined property handlers that need to match differently under `/i` must change to read the new boolean parameter passed it which is non-zero if case-insensitive matching is in effect; 0 if not. See *"User-Defined Character Properties" in perluniprops*.

### regex: \p{} in pattern implies Unicode semantics

Now, a Unicode property match specified in the pattern will indicate that the pattern is meant for matching according to Unicode rules (e40e74f)

### add GvCV_set() and GvGP_set() macros and change GvGP()

This allows a future commit to eliminate some backref magic between GV and CVs, which will require complete control over assignment to the gp_cv slot.

If you've been using GvGP() in lvalue context this change will break your code, you should use GvGP_set() instead. (c43ae56)

### _swash_inversion_hash is no longer exported as part of the API

This function shouldn't be called from XS code. (4c2e113)

### Unreferenced objects in global destruction

The fix for [perl #36347], which made sure that destructors were called on unreferenced objects, broke the tests for three CPAN modules, which apparently rely on the bug.

To provide more time for fixing them (as this is such a minor bug), we have reverted the fix until after perl 5.14.0.

This resolves [perl #82542] and other related tickets.

### close on shared pipes

The `close` function no longer waits for the child process to exit if the underlying file descriptor is still in use by another thread, to avoid deadlocks. It returns true in such cases.

## Deprecations

Deprecated Modules

The following modules will be removed from the core distribution in a future release, and should be installed from CPAN instead. Distributions on CPAN which require these should add them to their prerequisites. The core versions of these modules warnings will issue a deprecation warning.

If you ship a packaged version of Perl, either alone or as part of a larger system, then you should carefully consider the repercussions of core module deprecations. You may want to consider shipping your default build of Perl with packages for some or all deprecated modules which install into `vendor` or `site` perl library directories. This will inhibit the deprecation warnings.

Alternatively, you may want to consider patching *lib/deprecate.pm* to provide deprecation warnings specific to your packaging system or distribution of Perl, consistent with how your packaging system or distribution manages a staged transition from a release where the installation of a single package provides the given functionality, to a later release where the system administrator needs to know to install multiple packages to get that same functionality.

You can silence these deprecation warnings by installing the modules in question from CPAN. To install the latest version of all of them, just install `Task::Deprecations::5_14`.

*Devel::DProf*

We strongly recommend that you install and used *Devel::NYTProf* in preference, as it offers significantly improved profiling and reporting.

### User-defined case-mapping

This feature is being deprecated due to its many issues, as documented in *"User-Defined Case Mappings (for serious hackers only)" in perlunicode*. It is planned to remove this feature in Perl 5.16. A CPAN module providing improved functionality is being prepared for release by the time 5.14 is.

## Modules and Pragmata

### New Modules and Pragmata

- `CPAN::Meta` version 2.110440 has been added as a dual-life module. It provides a standard library to read, interpret and write CPAN distribution metadata files (e.g. META.json and META.yml) which describes a distribution, its contents, and the requirements for building it and installing it. The latest CPAN distribution metadata specification is included as `CPAN::Meta::Spec` and notes on changes in the specification over time are given in `CPAN::Meta::History`.
- `Version::Requirements` version 0.101020 has been added as a dual-life module. It provides a standard library to model and manipulates module prerequisites and version constraints as defined in the *CPAN::Meta::Spec*.

### Updated Modules and Pragmata

- `B` has been upgraded from version 1.27 to 1.28.
- `Carp` has been upgraded from version 1.19 to 1.20.

  [perl #82854] It now avoids using regular expressions that cause perl to load its Unicode tables, in order to avoid the 'BEGIN not safe after errors' error that will ensue if there has been a syntax error.
- `CGI` has been upgraded from version 3.51 to 3.52
- `CPAN` has been upgraded from version 1.94_64 to 1.94_65

  Includes support for META.json and MYMETA.json.
- `CPANPLUS` has been upgraded from version 0.9011 to 0.9101

  Includes support for META.json and MYMETA.json and a change to using Digest::SHA for CPAN checksums.
- `deprecate` has been upgraded from version 0.01 to 0.02.
- `diagnostics` has been upgraded from version 1.21 to 1.22.

  It now renders pod links slightly better, and has been taught to find descriptions for messages that share their descriptions with other messages.
- `Devel::DProf` has been upgraded from version 20080331.00 to 20110217.00.

  Merely loading `Devel::DProf` now no longer triggers profiling to start. `use Devel::DProf` and `perl -d:DProf ...` still behave as before and start the profiler.

  NOTE: `Devel::DProf` is deprecated and will be removed from a future version of Perl. We strongly recommend that you install and use *Devel::NYTProf* instead, as it offers significantly improved profiling and reporting.
- `DynaLoader` has been upgraded from version 1.12 to 1.13.

  [perl #84358] It no longer inherits from AutoLoader; hence it no longer produces weird error messages for unsuccessful method calls on classes that inherit from DynaLoader.
- `IO::Select` has been upgraded from version 1.17 to 1.18.

  It now allows IO::Handle objects (and objects in derived classes) to be removed from an IO::Select set even if the underlying file descriptor is closed or invalid.

- `IPC::Cmd` has been upgraded from version 0.68 to 0.70
- `HTTP::Tiny` has been upgraded from version 0.009 to 0.010
- `Math::BigInt` has been upgraded from version 1.99_04 to 1.992.
- `Module::Build` has been upgraded from version 0.3607 to 0.37_05.

  A notable change is the deprecation of several modules. Module::Build::Version has been deprecated and Module::Build now relies directly upon *version*. Module::Build::ModuleInfo has been deprecated in favor of a standalone copy of it called *Module::Metadata*. Module::Build::YAML has been deprecated in favor of *CPAN::Meta::YAML*.

  Module::Build now also generates META.json and MYMETA.json files in accordance with version 2 of the CPAN distribution metadata specification, *CPAN::Meta::Spec*. The older format META.yml and MYMETA.yml files are still generated, as well.
- `Module::Load::Conditional` has been upgraded from version 0.40 to 0.44
- `Module::Metadata` has been upgraded from version 1.000003 to 1.000004.
- `overload` has been upgraded from version 1.12 to 1.13.

  The documentation has greatly improved. See *Documentation* below.
- `Parse::CPAN::Meta` has been upgraded from version 1.40 to 1.4401.

  The latest Parse::CPAN::Meta can now read YAML or JSON files using *CPAN::Meta::YAML* and *JSON::PP*, which are now part of the Perl core.
- `re` has been upgraded from version 0.16 to 0.17.

  It now supports the double-a flag: `use re '/aa';`

  The `regmust` function used to crash when called on a regular expression belonging to a pluggable engine. Now it has been disabled for those.

  `regmust` no longer leaks memory.
- `Term::UI` has been upgraded from version 0.24 to 0.26
- `Unicode::Collate` has been upgraded from version 0.68 to 0.72

  This also sees the switch from using the pure-perl version of this module to the XS version.`
- `VMS::DCLsym` has been upgraded from version 1.04 to 1.05.

  Two bugs have been fixed [perl #84086]:

  The symbol table name was lost when tying a hash, due to a thinko in `TIEHASH`. The result was that all tied hashes interacted with the local symbol table.

  Unless a symbol table name had been explicitly specified in the call to the constructor, querying the special key ':LOCAL' failed to identify objects connected to the local symbol table.
- Added new function `Unicode::UCD::num()`. This function will return the numeric value of the string passed it; `undef` if the string in its entirety has no safe numeric value.

  To be safe, a string must be a single character which has a numeric value, or consist entirely of characters that match \d, coming from the same Unicode block of digits. Thus, a mix of Bengali and Western digits would be considered unsafe, as well as a mix of half- and full-width digits, but strings consisting entirely of Devanagari digits or of "Mathematical Bold" digits would would be safe.
- `CPAN` has been upgraded from version 1.94_63 to 1.94_64.

## Documentation

### Changes to Existing Documentation

#### overload

- *overload*'s documentation has practically undergone a rewrite. It is now much more straightforward and clear.

#### perlhack and perlrepository

- The *perlhack* and perlrepository documents have been heavily edited and split up into several new documents.

  The *perlhack* document is now much shorter, and focuses on the Perl 5 development process and submitting patches to Perl. The technical content has been moved to several new documents, *perlsource*, *perlinterp*, *perlhacktut*, and *perlhacktips*. This technical content has only been lightly edited.

  The perlrepository document has been renamed to *perlgit*. This new document is just a how-to on using git with the Perl source code. Any other content that used to be in perlrepository has been moved to perlhack.

#### perlfunc

- The documentation for the `map` function now contains more examples, see **perldoc -f map** (f947627)

#### perlfaq4

- Examples in *perlfaq4* have been updated to show the use of *Time::Piece*. (9243591)

#### Miscellaneous

- Many POD related RT bugs and other issues which are too numerous to enumerate have been solved by Michael Stevens.

## 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

"\b{" is deprecated; use "\b\{" instead

"\B{" is deprecated; use "\B\{" instead

Use of an unescaped "{" immediately following a `\b` or `\B` is now deprecated so as to reserve its use for Perl itself in a future release.

regcomp: Add warning if \p is used under locale. (fb2e24c)

`\p` implies Unicode matching rules, which are likely going to be different than the locale's.

panic: gp_free failed to free glob pointer - something is repeatedly re-creating entries

This new error is triggered if a destructor called on an object in a typeglob that is being freed creates a new typeglob entry containing an object with a destructor that creates a new entry containing an object....

refcnt: fd %d%s

This new error only occurs if a internal consistency check fails when a pipe is about to be closed.

### Changes to Existing Diagnostics

- The warning message about regex unrecognized escapes passed through is changed to include any literal '{' following the 2-char escape. e.g., "\q{" will include the { in the message as part of the escape (216bfc0).

- `binmode $fh, ':scalar'` no longer warns (8250589)

  Perl will now no longer produce this warning:

  ```
  $ perl -we 'open my $f, ">", \my $x; binmode $f, "scalar"'
  Use of uninitialized value in binmode at -e line 1.
  ```

## Utility Changes

### perlbug

- [perl #82996] Use the user's from address as return-path in perlbug

  Many systems these days don't have a valid Internet domain name and perlbug@perl.org does not accept email with a return-path that does not resolve. Therefore pass the user's address to sendmail so it's less likely to get stuck in a mail queue somewhere. (019cfd2)

## Configuration and Compilation

- make reg_eval_scope.t TODOs consistently fail (daaf7ac)

  Some of the TODO tests in reg_eval_scope.t spuriously passed under non-threaded builds. Make the tests harder so they always fail.

  Since one of the key bugs in (?{..}) is the trashing of the parent pad, add some extra lexical vars to the parent scope and check they're still there at the end.

- Stop EU::CBuilder's tests from failing in parallel (cbf59d5)

  It used to use the same paths for temporary files in all tests. This blew up randomly when the tests were run in parallel.

## Testing

- *porting/FindExt.t* now skips all tests on a static (-Uusedl) build of perl.
- *porting/FindExt.t* now passes on non-Win32 platforms when some extensions are built statically.

## Platform Support

### Platform-Specific Notes

Windows

- The `test-prep` build target now depends on *pod/perltoc.pod* to allow the *t/porting/buildtoc.t* test to run successfully.

MirBSD

- [perl #82988] Skip hanging taint.t test on MirBSD 10 (1fb83d0)

  Skip a hanging test under MirBSD that was already being skipped under OpenBSD.

- Previously if you build perl with a shared libperl.so on MirBSD (the default config), it will work up to the installation; however, once installed, it will be unable to find libperl. Treat path handling like in the other BSD dialects.

## Internal Changes

- Fix harmless invalid read in Perl_re_compile() (f6d9469)

  [perl #2460] described a case where electric fence reported an invalid read. This could be reproduced under valgrind with blead and -e'/x/', but only on a non-debugging build.

  This was because it was checking for certain pairs of nodes (e.g. BOL + END) and wasn't allowing for EXACT nodes, which have the string at the next node position when using a naive NEXTOPER(first). In the non-debugging build, the nodes aren't initialised to zero, and a 1-char EXACT node isn't long enough to spill into the type field of the "next node".

  Fix this by only using NEXTOPER(first) when we know the first node is kosher.

- Break out the generated function Perl_keywords() into *keywords.c*, a new file. (26ea9e1)

  As it and Perl_yylex() both need FEATURE_IS_ENABLED, feature_is_enabled() is no longer static, and the two macro definitions move from toke.c to perl.h

  Previously, one had to cut and paste the output of perl_keywords.pl into the middle of toke.c, and it was not clear that it was generated code.

- A lot of tests have been ported from Test to Test::More, e.g. in 3842ad6.

- Increase default PerlIO buffer size. (b83080d)

  The previous default size of a PerlIO buffer (4096 bytes) has been increased to the larger of 8192 bytes and your local BUFSIZ. Benchmarks show that doubling this decade-old default increases read and write performance in the neighborhood of 25% to 50% when using the default layers of perlio on top of unix. To choose a non-default size, such as to get back the old value or to obtain and even larger value, configure with:

  ```
      ./Configure -Accflags=-DPERLIOBUF_DEFAULT_BUFSIZ=N
  ```

  where N is the desired size in bytes; it should probably be a multiple of your page size.

## Selected Bug Fixes

- A Unicode `\p{}` property match in a regular expression pattern will now force Unicode rules for the rest of the regular expression

- [perl #38456] binmode FH, ":crlf" only modifies top crlf layer (7826b36)

  When pushed on top of the stack, crlf will no longer enable crlf layers lower in the stack. This will prevent unexpected results.

- Fix 'raw' layer for RT #80764 (ecfd064)

  Made a ':raw' open do what it advertises to do (first open the file, then binmode it), instead of leaving off the top layer.

- Use PerlIOBase_open for pop, utf8 and bytes layers (c0888ac)

  Three of Perl's builtin PerlIO layers (`:pop`, `:utf8` and `:bytes`) didn't allow stacking when opening a file. For example this:

  ```
      open FH, '>:pop:perlio', 'some.file' or die $!;
  ```

  Would throw an error: "Invalid argument". This has been fixed in this release.

- An issue present since 5.13.1, where s/A/B/ with A utf8 and B non-utf8, could cause corruption or segfaults has been fixed. (c95ca9b)

- String evals will no longer fail after 2 billion scopes have been compiled (d1bfb64, 2df5bdd, 0d311cd and 6012dc8)

- [perl #81750] When strict 'refs' mode is off, `%{...}` in rvalue context returns `undef` if its argument is undefined. An optimisation introduced in perl 5.12.0 to make `keys %{...}` faster when used as a boolean did not take this into account, causing `keys %{+undef}` (and `keys %$foo` when `$foo` is undefined) to be an error, which it should only be in strict mode.

- [perl #83194] Combining the vector (%v) flag and dynamic precision would cause sprintf to confuse the order of its arguments, making it treat the string as the precision and vice versa.

- [perl #77692] Sometimes the UTF8 length cache would not be reset on a value returned by substr, causing `length(substr($uni_string,...))` to give wrong answers. With `${^UTF8CACHE}` set to -1, it would produce a 'panic' error message, too.

- During the restoration of a localised typeglob on scope exit, any destructors called as a result would be able to see the typeglob in an inconsistent state, containing freed entries, which could result in a crash. This would affect code like this:

```
local *@;
eval { die bless [] }; # puts an object in $@
sub DESTROY {
  local $@; # boom
}
```

Now the glob entries are cleared before any destructors are called. This also means that destructors can vivify entries in the glob. So perl tries again and, if the entries are re-created too many times, dies with a 'panic: gp_free...' error message.

- [perl #78494] When pipes are shared between threads, the `close` function (and any implicit close, such as on thread exit) no longer blocks.
- Several contexts no longer allow a Unicode character to begin a word that should never begin words, for an example an accent that must follow another character previously could precede all other characters.
- Case insensitive matching in regular expressions compiled under `use locale` now works much more sanely when the pattern and/or target string are encoded in UTF-8. Previously, under these conditions the localeness was completely lost. Now, code points above 255 are treated as Unicode, but code points between 0 and 255 are treated using the current locale rules, regardless of whether the pattern or string are encoded in UTF-8. The few case insensitive matches that cross the 255/256 boundary are not allowed. For example, 0xFF does not caselessly match the character at 0x178, LATIN CAPITAL LETTER Y WITH DIAERESIS, because 0xFF may not be LATIN SMALL LETTER Y in the current locale, and Perl has no way of knowing if that character even exists in the locale, much less what code point it is.

## Acknowledgements

Perl 5.13.10 represents approximately one month of development since Perl 5.13.9 and contains approximately 63000 lines of changes across 609 files from 38 authors and committers:

Abigail, Alexander Hartmaier, brian d foy, Charles Bailey, Chip Salzenberg, Chris 'BinGOs' Williams, Craig A. Berry, Curtis Jewell, Dave Rolsky, David Golden, David Leadbeater, David Mitchell, David Wheeler, Father Chrysostomos, Florian Ragwitz, Franz Fasching, George Greer, H.Merijn Brand, Hongwen Qiu, Hugo van der Sanden, Jay Hannah, Jesse Vincent, Karl Williamson, Larwan Berke, Leon Timmermans, Michael Breen, Michael Stevens, Nicholas Clark, Noirin Shirley, Paul Evans, Peter John Acklam, Ricardo Signes, Robin Barker, Steven Schubiger, Tom Christiansen, Tony Cook, ZsbÃ¡n Ambrus 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.