

NAME

perl5134delta - what is new for perl v5.13.4

DESCRIPTION

This document describes differences between the 5.13.4 release and the 5.13.3 release.

If you are upgrading from an earlier release such as 5.13.2, first read *perl5133delta*, which describes differences between 5.13.2 and 5.13.3.

Core Enhancements

srand() now returns the seed

This allows programs that need to have repeatable results to not have to come up with their own seed generating mechanism. Instead, they can use `srand()` and somehow stash the return for future use. Typical is a test program which has too many combinations to test comprehensively in the time available to it each run. It can test a random subset each time, and should there be a failure, log the seed used for that run so that it can later be used to reproduce the exact results.

\N{name} and charnames enhancements

`\N{}`, `charnames::vianame`, `charnames::viacode` now know about every character in Unicode. Previously, they didn't know about the Hangul syllables nor a number of CJK (Chinese/Japanese/Korean) characters.

Incompatible Changes

Declare API incompatibility between blead releases

Only stable releases (5.10.x, 5.12.x, 5.14.x, ...) guarantee binary compatibility with each other, while blead releases (5.13.x, 5.15.x, ...) often break this compatibility. However, prior to perl 5.13.4, all blead releases had the same `PERL_API_REVISION`, `PERL_API_VERSION`, and `PERL_API_SUBVERSION`, effectively declaring them as binary compatible, which they weren't. From now on, blead releases will have a `PERL_API_SUBVERSION` equal to their `PERL_SUBVERSION`, explicitly marking them as incompatible with each other.

Maintenance releases of stable perl versions will continue to make no intentionally incompatible API changes.

Check API compatibility when loading XS modules

When perl's API changes in incompatible ways (which usually happens between every major release), XS modules compiled for previous versions of perl will not work anymore. They will need to be recompiled against the new perl.

In order to ensure that modules are recompiled, and to prevent users from accidentally loading modules compiled for old perls into newer ones, the `XS_APIVERSION_BOOTCHECK` macro has been added. That macro, which is called when loading every newly compiled extension, compares the API version of the running perl with the version a module has been compiled for and raises an exception if they don't match.

Binary Incompatible with all previous Perls

Some bit fields have been reordered; therefore, this release will not be binary compatible with any previous Perl release.

Change in the parsing of certain prototypes

Functions declared with the following prototypes now behave correctly as unary functions:

- `*`
- `\sigil`
- `\[...]`

- `;$`
- `;*`
- `;\sigil`
- `;\[...]`

Due to this bug fix, functions using the `(*)`, `(;$)` and `(;*)` prototypes are parsed with higher precedence than before. So in the following example:

```
sub foo($);
foo $a < $b;
```

the second line is now parsed correctly as `foo($a) < $b`, rather than `foo($a < $b)`. This happens when one of these operators is used in an unparenthesised argument:

```
< > <= >= lt gt le ge
== != <=> eq ne cmp ~~
&
| ^
&&
|| //
.. ...
?:
= += -= *= etc.
```

Deprecations

List assignment to `$[`

After assignment to `$[` has been deprecated and started to give warnings in perl version 5.12.0, this version of perl also starts to emit a warning when assigning to `$[` in list context. This fixes an oversight in 5.12.0.

Performance Enhancements

- Make string appending 100 times faster
When doing a lot of string appending, perl could end up allocating a lot more memory than needed in a very inefficient way, if perl was configured to use the system's `malloc` implementation instead of its own.
`sv_grow`, which is what's being used to allocate more memory if necessary when appending to a string, has now been taught how to round up the memory it requests to a certain geometric progression, making it much faster on certain platforms and configurations. On Win32, it's now about 100 times faster.
- For weak references, the common case of just a single weak reference per referent has been optimised to reduce the storage required. In this case it saves the equivalent of one small perl array per referent.
- `XPV`, `XPVIV`, and `XPVNV` now only allocate the parts of the `SV` body they actually use, saving some space.

Modules and Pragmata

New Modules and Pragmata

This release does not introduce any new modules or pragmata.

Updated Modules and Pragmata

Archive::Tar

Upgraded from version 1.64 to 1.68.

Among other things, the new version adds a new option to `ptar` to allow safe creation of tarballs without world-writable files on Windows, allowing those archives to be uploaded to CPAN.

B::Lint

Upgraded from version 1.11 to 1.12.

Carp

Upgraded from version 1.16 to 1.18.

Carp now detects incomplete *caller()* overrides and avoids using bogus `@DB::args`. To provide backtraces, *Carp* relies on particular behaviour of the *caller* built-in. *Carp* now detects if other code has overridden this with an incomplete implementation, and modifies its backtrace accordingly. Previously incomplete overrides would cause incorrect values in backtraces (best case), or obscure fatal errors (worst case)

This fixes certain cases of Bizarre copy of ARRAY caused by modules overriding *caller()* incorrectly.

Compress::Raw::Bzip2

Upgraded from version 2.027 to 2.030.

Compress::Raw::Zlib

Upgraded from version 2.027 to 2.030.

File::Spec

Upgraded from version 3.31 to 3.31_01.

Various issues in *File::Spec::VMS* have been fixed.

I18N::Langinfo

Upgraded from version 0.03 to 0.04.

langinfo() now defaults to using `$_` if there is no argument given, just like the documentation always claimed it did.

IO::Compress

Upgraded from version 2.027 to 2.030.

Module::CoreList

Upgraded from version 2.36 to 2.37.

Besides listing the updated core modules of this release, it also stops listing the *Filespec* module. That module never existed in core. The scripts generating *Module::CoreList* confused it with *VMS::Filespec*, which actually is a core module, since the time of perl 5.8.7.

Test::Harness

Upgraded from version 3.21 to 3.22.

Test::Simple

Upgraded from version 0.94 to 0.96.

Among many other things, subtests without a `plan` or `no_plan` now have an implicit `done_testing()` added to them.

Unicode::Collate

Upgraded from version 0.53 to 0.56.

Among other things, it is now using UCA Revision 20 (based on Unicode 5.2.0) and supports a couple of new locales.

feature

Upgraded from version 1.17 to 1.18.

Removed Modules and Pragmata

This release does not remove any modules or pragmata.

Documentation

Changes to Existing Documentation

perldiag

- The following existing diagnostics are now documented:
 - *Ambiguous use of %c resolved as operator %c*
 - *Ambiguous use of %c{%s} resolved to %c%s*
 - *Ambiguous use of %c{%s%s} resolved to %c%s%s*
 - *Ambiguous use of -%s resolved as -&%s()*
 - *Invalid strict version format (%s)*
 - *Invalid version format (%s)*
 - *Invalid version object*

perlport

- Documented a *limitation* of *alarm()* on Win32.

perlre

- Minor fix to a multiple scalar match example.

Configuration and Compilation

- Compatibility with C++ compilers has been improved.
- On compilers that support it, `-Wwrite-strings` is now added to `cflags` by default.

Testing

- *t/op/print.t* has been added to test implicit printing of `$_`.
- *t/io/errnosig.t* has been added to test for restoration of `$!` when leaving signal handlers.
- *t/op/tie_fetch_count.t* has been added to see if `FETCH` is only called once on tied variables.
- *lib/Tie/ExtraHash.t* has been added to make sure the, previously untested, `Tie::ExtraHash` keeps working.
- *t/re/overload.t* has been added to test against string corruption in pattern matches on overloaded objects. This is a TODO test.

Platform Support

Platform-Specific Notes

Win32

- Fixed a possible hang in *t/op/readline.t*.

- Fixed build process for SDK2003SP1 compilers.
- When using old 32-bit compilers, the define `_USE_32BIT_TIME_T` will now be set in `$Config{ccflags}`. This improves portability when compiling XS extensions using new compilers, but for a perl compiled with old 32-bit compilers.

Internal Changes

Removed `PERL_POLLUTE`

The option to define `PERL_POLLUTE` to expose older 5.005 symbols for backwards compatibility has been removed. It's use was always discouraged, and MakeMaker contains a more specific escape hatch:

```
perl Makefile.PL POLLUTE=1
```

This can be used for modules that have not been upgraded to 5.6 naming conventions (and really should be completely obsolete by now).

Added `PERL_STATIC_INLINE`

The `PERL_STATIC_INLINE` define has been added to provide the best-guess incantation to use for static inline functions, if the C compiler supports C99-style static inline. If it doesn't, it'll give a plain `static`.

`HAS_STATIC_INLINE` can be used to check if the compiler actually supports inline functions.

Selected Bug Fixes

- A possible memory leak when using `caller()` to set `@DB::args` has been fixed.
- Several memory leaks when loading XS modules were fixed.
- A panic in the regular expression optimizer has been fixed (RT#75762).
- Assignments to lvalue subroutines now honor copy-on-write behavior again, which has been broken since version 5.10.0 (RT#75656).
- Assignments to glob copies now behave just like assignments to regular globs (RT#1804).
- Within signal handlers, `$!` is now implicitly localized.
- `readline` now honors `<>` overloading on tied arguments.
- `substr()`, `pos()`, `keys()`, and `vec()` could, when used in combination with lvalues, result in leaking the scalar value they operate on, and cause its destruction to happen too late. This has now been fixed.
- Building with `PERL_GLOBAL_STRUCT`, which has been broken accidentally in 5.13.3, now works again.

Known Problems

- The changes in `substr()` broke `HTML::Parser <= 3.66`. A fixed `HTML::Parser` is available as version 3.67 on CPAN.
- The changes in prototype handling break `Switch`. A patch has been sent upstream and will hopefully appear on CPAN soon.

Acknowledgements

Perl 5.13.4 represents approximately one month of development since Perl 5.13.3, and contains 91,200 lines of changes across 436 files from 34 authors and committers.

Thank you to the following for contributing to this release:

Abigail, Andy Armstrong, Andy Dougherty, Chas. Owens, Chip Salzenberg, Chris 'BinGOs' Williams,

