NAME

perl5138delta - what is new for perl v5.13.8

DESCRIPTION

This document describes differences between the 5.13.7 release and the 5.13.8 release.

If you are upgrading from an earlier release such as 5.13.6, first read *perl5137delta*, which describes differences between 5.13.6 and 5.13.7.

Core Enhancements

-d:-foo calls Devel::foo::unimport

The syntax `-d:`***foo*** was extended in 5.6.1 to make `-d:`***foo*=*bar*** equivalent to `-MDevel::foo=bar`, which expands internally to `use Devel::foo 'bar';`. *perl* now allows prefixing the module name with `-`, with the same semantics as `-M`, *i.e.*

`-d:-foo`

Equivalent to `-M-Devel::foo`, expands to `no Devel::foo;`, calls `Devel::foo->unimport()` if the method exists.

`-d:-foo=bar`

Equivalent to `-M-Devel::foo=bar`, expands to `no Devel::foo 'bar';`, calls `Devel::foo->unimport('bar')` if the method exists.

This is particularly useful to suppresses the default actions of a `Devel::*` module's `import` method whilst still loading it for debugging.

Filehandle method calls load IO::File on demand

When a method call on a filehandle would die because the method cannot be resolved, and *IO::File* has not been loaded, Perl now loads *IO::File* via `require` and attempts method resolution again:

```
open my $fh, ">", $file;
$fh->binmode(":raw");     # loads IO::File and succeeds
```

This also works for globs like STDOUT, STDERR and STDIN:

```
STDOUT->autoflush(1);
```

Because this on-demand load only happens if method resolution fails, the legacy approach of manually loading an *IO::File* parent class for partial method support still works as expected:

```
use IO::Handle;
open my $fh, ">", $file;
$fh->autoflush(1);        # IO::File not loaded
```

Full functionality for use feature 'unicode_strings'

This release provides full functionality for `use feature 'unicode_strings'`. Under its scope, all string operations executed and regular expressions compiled (even if executed outside its scope) have Unicode semantics. See *feature*.

This feature avoids most forms of the "Unicode Bug" (See *"The "Unicode Bug"" in perlunicode* for details.) If there is a possibility that your code will process Unicode strings, you are **strongly** encouraged to use this subpragma to avoid nasty surprises.

The availability of this should strongly affect the whole tone of various documents, such as *perlunicode* and *perluniintro*, but this work has not been done yet.

Exception Handling Backcompat Hack

When an exception is thrown in an `eval BLOCK`, `$@` is now set before unwinding, as well as being set after unwinding as the eval block exits. This early setting supports code that has historically treated `$@` during unwinding as an indicator of whether the unwinding was due to an exception. These modules had been broken by 5.13.1's change from setting `$@` early to setting it late. This double setting arrangement is a stopgap until the reason for unwinding can be made properly introspectable. `$@` has never been a reliable indicator of the reason for unwinding.

printf-like functions understand post-1980 size modifiers

Perl's printf and sprintf operators, and Perl's internal printf replacement function, now understand the C90 size modifiers "hh" (`char`), "z" (`size_t`), and "t" (`ptrdiff_t`). Also, when compiled with a C99 compiler, Perl now understands the size modifier "j" (`intmax_t`).

So, for example, on any modern machine, `sprintf('%hhd', 257)` returns '1'.

DTrace probes now include package name

The DTrace probes now include an additional argument (`arg3`) which contains the package the subroutine being entered or left was compiled in.

For example using the following DTrace script:

```
perl$target:::sub-entry
{
    printf("%s::%s\n", copyinstr(arg0), copyinstr(arg3));
}
```

and then running:

```
perl -e'sub test { }; test'
```

DTrace will print:

```
main::test
```

Stacked labels

Multiple statement labels can now appear before a single statement.

Incompatible Changes

:= is now a syntax error

Previously `my $pi := 4;` was exactly equivalent to `my $pi : = 4;`, with the `:` being treated as the start of an attribute list, ending before the `=`. The use of `:=` to mean `: =` was deprecated in 5.12.0, and is now a syntax error. This will allow the future use of `:=` as a new token.

We find no Perl 5 code on CPAN using this construction, outside the core's tests for it, so we believe that this change will have very little impact on real-world codebases.

If it is absolutely necessary to have empty attribute lists (for example, because of a code generator) then avoid the error by adding a space before the `=`.

Run-time code block in regular expressions

Code blocks in regular expressions (`(?{...})` and `(??{...})`) used not to inherit any pragmata (strict, warnings, etc.) if the regular expression was compiled at run time as happens in cases like these two:

```
use re 'eval';
$foo =~ $bar; # when $bar contains (?{...})
$foo =~ /$bar(?{ $finished = 1 })/;
```

This was a bug, which has now been fixed. But it has the potential to break any code that was relying on this bug.

Deprecations

?PATTERN? is deprecated

`?PATTERN?` (without the initial m) has been deprecated and now produces a warning. This is to allow future use of ? in new operators. The match-once functionality is still available in the form of `m?PATTERN?`.

sv_compile_2op() is now deprecated

The `sv_compile_2op()` API function is now deprecated. Searches suggest that nothing on CPAN is using it, so this should have zero impact.

It attempted to provide an API to compile code down to an optree, but failed to bind correctly to lexicals in the enclosing scope. It's not possible to fix this problem within the constraints of its parameters and return value.

Tie functions on scalars holding typeglobs

Calling a tie function (`tie`, `tied`, `untie`) with a scalar argument acts on a file handle if the scalar happens to hold a typeglob.

This is a long-standing bug that will be removed in Perl 5.16, as there is currently no way to tie the scalar itself when it holds a typeglob, and no way to untie a scalar that has had a typeglob assigned to it.

This bug was fixed in 5.13.7 but, because of the breakage it caused, the fix has been reverted. Now there is a deprecation warning whenever a tie function is used on a handle without an explicit `*`.

Modules and Pragmata

Updated Modules and Pragmata

- `Archive::Tar` has been upgraded from version 1.72 to 1.74.

  Skip extracting pax extended headers.

- `autodie` has been upgraded from version 2.10 to 2.1001.

  Test fix in blead for VMS.

- `B` has been upgraded from version 1.26 to 1.27.

  Avoid compiler warnings.

- `B::Concise` has been upgraded from version 0.81 to 0.82.

  It no longer produces mangled output with the `-tree` option *[perl #80632]*.

- `B::Deparse` has been upgraded from version 1.01 to 1.02.

  Test improvements.

- `Cwd` has been upgraded from version 3.34 to 3.35.

  Avoid compiler warnings.

- `Data::Dumper` has been upgraded from version 2.130_01 to 2.130_02.

  Avoid compiler warnings.

- `Devel::Peek` has been upgraded from version 1.05 to 1.06.

  Avoid compiler warnings.

  Test improvements.

- `Devel::SelfStubber` has been upgraded from version 1.03 to 1.05.

Whitespace changes.

- `Digest::SHA` has been upgraded from 5.48 to 5.50.

  `shasum` now more closely mimics `sha1sum`/`md5sum`.

  `Addfile` accepts all POSIX filenames.

- `Dumpvalue` has been upgraded from version 1.14 to 1.15.

  Test improvements.

- `DynaLoader` has been upgraded from version 1.11 to 1.12.

  Remove obsolete RCS keywords.

- `Env` has been upgraded from version 1.01 to 1.02.

  Test improvements.

- `ExtUtils::CBuilder` has been upgraded from 0.2703 to 0.280201.

  Handle C and C++ compilers separately.

  Preserves exit status on VMS.

  Test improvements.

- `ExtUtils::Constant::Utils` has been upgraded from 0.02 to 0.03.

  Refactoring and fixing of backcompat code, preparing for resynchronisation with CPAN.

- `ExtUtils::Embed` has been upgraded from 1.29 to 1.30.

  Remove obsolete RCS keywords.

- `ExtUtils::ParseXS` has been upgraded from 2.2207 to 2.2208.

  Avoid compiler warnings.

- `Fcntl` has been upgraded from 1.10 to 1.11.

  Avoid compiler warnings.

  Test improvements.

- `feature` has been upgraded from 1.18 to 1.19.

  Documentation and test updates for the `unicode_strings` feature. See *Full functionality for `use feature 'unicode_strings'`*.

- `File::CheckTree` has been upgraded from 4.4 to 4.41.

  Whitespace changes.

- `File::Glob` has been upgraded from 1.10 to 1.11.

  Avoid compiler warnings.

  Test improvements.

- `GDBM_File` has been upgraded from 1.12 to 1.13.

  Test improvements.

  Remove obsolete RCS keywords.

- `Hash::Util::FieldHash` has been upgraded from 1.06 to 1.07.

  Avoid compiler warnings.

- `I18N::Collate` has been upgraded from 1.01 to 1.02.

  Whitespace changes.

  Test improvements.

- `if` has been upgraded from 0.06 to 0.0601.

  Test improvements.

- `IO` has been upgraded from 1.25_02 to 1.25_03.

  Avoid compiler warnings.

- `IPC::Cmd` has been upgraded from 0.64 to 0.66.

  Resolves an issue with splitting Win32 command lines.

  Documentation enhancements.

- `IPC::Open3` has been upgraded from 1.07 to 1.08.

  Remove obsolete RCS keywords.

  Test improvements.

- `Locale::Codes` has been upgraded from version 3.14 to 3.15.

  Adds some codes.

- `Math::BigInt` has been upgraded from 1.99_01 to 1.99_02.

  Documentation and comment spelling fixes.

- `Memoize` has been upgraded from version 1.01_03 to 1.02.

  Remove obsolete RCS keywords.

  Whitespace changes.

- `MIME::Base64` has been upgraded from 3.10 to 3.13.

  Now provides `encode_base64url` and `decode_base64url` functions to process the base64 scheme for "URL applications".

- `mro` has been upgraded from version 1.05 to 1.06.

  `next::method` *et al.* now take into account that every class inherits from UNIVERSAL *[perl #68654]*.

- `NDBM_File` has been upgraded from 1.10 to 1.11.

  Remove obsolete RCS keywords.

  Test improvements.

- `Net::Ping` has been upgraded from 2.36 to 2.37.

  Remove obsolete RCS keywords.

- `ODBM_File` has been upgraded from 1.09 to 1.10.

  Remove obsolete RCS keywords.

  Test improvements.

- `Opcode` has been upgraded from 1.17 to 1.18.

  Avoid compiler warnings.

  Test improvements.

- `overload` has been upgraded from 1.11 to 1.12.

  Avoid a taint problem in use of sprintf.

  Test asymmetric fallback cases *[perl #71286]*.

- `PerlIO::encoding` has been upgraded from 0.13 to 0.14.

  Avoid compiler warnings.

Remove obsolete RCS keywords.

Test improvements.

- `PerlIO::scalar` has been upgraded from 0.10 to 0.11.

  A `read` after a `seek` beyond the end of the string no longer thinks it has data to read *[perl #78716]*.

  Avoid compiler warnings.

- `PerlIO::via` has been upgraded from 0.10 to 0.11.

  Avoid compiler warnings.

- `POSIX` has been upgraded from 1.22 to 1.23.

  Avoid compiler warnings.

- `re` has been upgraded from 0.14 to 0.15.

  Enforce that `/d`, `/u`, and `/l` are mutually exclusive.

- `SDBM_File` has been upgraded from 1.08 to 1.09.

  Avoid compiler warnings.

  Remove obsolete RCS keywords.

  Test improvements.

- `Socket` has been upgraded from 1.91 to 1.92.

  It has several new functions for handling IPv6 addresses.

- `Storable` has been upgraded from 2.24 to 2.25.

  This adds support for serialising code references that contain UTF-8 strings correctly. The Storable minor version number changed as a result, meaning that Storable users who set `$Storable::accept_future_minor` to a `FALSE` value will see errors (see *"FORWARD COMPATIBILITY" in Storable* for more details).

  Freezing no longer gets confused if the Perl stack gets reallocated during freezing *[perl #80074]*.

  Avoid compiler warnings.

- `threads` has been upgraded from 1.81_02 to 1.81_03.

  Avoid compiler warnings.

- `threads::shared` has been upgraded from 1.34 to 1.35.

  Avoid compiler warnings.

- `Time::HiRes` has been upgraded from 1.9721 to 1.9721_01.

  Build fix in blead for VMS.

- `Unicode::Collate` has been upgraded from 0.67 to 0.6801.

  Documentation clarification.

  Test improvements.

- `Unicode::Normalize` has been upgraded from 1.07 to 1.08.

  Avoid compiler warnings.

- `Unicode::UCD` has been upgraded from 0.29 to 0.30.

  Add info about named sequence alternatives.

  Don't use `CompositionExclusions.txt`.

- `version` has been upgraded from 0.82 to 0.86.

  Modify export logic for `is_strict` and `is_lax`.

  Various backcompat fixes.

- `Win32` has been upgraded from 0.39 to 0.41.

  Add several functions.

  Corrections to names returned by `Win32::GetOSName` and `Win32::GetOSDisplayName`.

- `XS::APItest` has been upgraded from 0.26 to 0.27.

  Test new API functions.

  Avoid compiler warnings.

Dual-life Modules and Pragmata

These modules were formerly distributed only in the Perl core distribution, and are now dual-lifed (meaning they are now also available separately on CPAN):

- `autouse`
- `Devel::SelfStubber`
- `Dumpvalue`
- `Env`
- `File::CheckTree`
- `I18N::Collate`

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

- There is a new "Closure prototype called" error *[perl #68560]*.

Changes to Existing Diagnostics

- The "Found = in conditional" warning that is emitted when a constant is assigned to a variable in a condition is now withheld if the constant is actually a subroutine or one generated by `use constant`, since the value of the constant may not be known at the time the program is written *[perl #77762]*.

Configuration and Compilation

- The `Encode` module can now (once again) be included in a static Perl build. The special-case handling for this situation got broken in Perl 5.11.0, and has now been repaired.

Testing

- Tests for `Fcntl`, `File::Glob`, `GDBM_File`, `IPC::Open3`, `NDBM_File`, `ODBM_File`, `Opcode`, `PerlIO::encoding`, `SDBM_File`, and `Storable` now use the *Test::More* framework.

Platform Support

Platform-Specific Notes

NetBSD

The NetBSD hints file has been changed to make the system's malloc the default.

Windows

The option to use an externally-supplied `crypt()`, or to build with no `crypt()` at all, has been removed. Perl supplies its own `crypt()` implementation for Windows, and the political situation that required this part of the distribution to sometimes be omitted is long gone.

Internal Changes

- The *`mg_findext()`* and *`sv_unmagicext()`* functions have been added to the API. They allow extension authors to find and remove magic attached to scalars based on both the magic type and the magic virtual table, similar to how `sv_magicext()` attaches magic of a certain type and with a given virtual table to a scalar. This eliminates the need for extensions to walk the list of `MAGIC` pointers of an `SV` to find the magic that belongs to them.

- The *`parse_fullexpr()`*, *`parse_listexpr()`*, *`parse_termexpr()`*, and *`parse_arithexpr()`* functions have been added to the API. They perform recursive-descent parsing of expressions at various precedence levels. They are expected to be used by syntax plugins.

Selected Bug Fixes

- `BEGIN {require 5.12.0}` now behaves as documented, rather than behaving identically to `use 5.12.0;`. Previously, `require` in a `BEGIN` block was erroneously executing the `use feature ':5.12.0'` and `use strict; use warnings;` behaviour, which only `use` was documented to provide *[perl #69050]*.

- `use 5.42` *[perl #69050]*, `use 6` and `no 5` no longer leak memory.

- `eval "BEGIN{die}"` no longer leaks memory on non-threaded builds.

- PerlIO no longer crashes when called recursively, e.g., from a signal handler. Now it just leaks memory *[perl #75556]*.

- Defining a constant with the same name as one of perl's special blocks (e.g., INIT) stopped working in 5.12.0, but has now been fixed *[perl #78634]*.

- A reference to a literal value used as a hash key (`$hash{\"foo"}`) used to be stringified, even if the hash was tied *[perl #79178]*.

- A closure containing an `if` statement followed by a constant or variable is no longer treated as a constant *[perl #63540]*.

- Calling a closure prototype (what is passed to an attribute handler for a closure) now results in a "Closure prototype called" error message instead of a crash *[perl #68560]*.

- A regular expression optimisation would sometimes cause a match with a `{n,m}` quantifier to fail when it should match *[perl #79152]*.

- What has become known as the "Unicode Bug" is mostly resolved in this release. Under `use feature 'unicode_strings'`, the internal storage format of a string no longer affects the external semantics. There are two known exceptions. User-defined case changing functions, which are planned to be deprecated in 5.14, require utf8-encoded strings to function; and the character `LATIN SMALL LETTER SHARP S` in regular expression case-insensitive matching has a somewhat different set of bugs depending on the internal storage format. Case-insensitive matching of all characters that have multi-character matches, as this one does, is problematical in Perl. *[perl #58182]*.

- Mentioning a read-only lexical variable from the enclosing scope in a string `eval` no longer causes the variable to become writable *[perl #19135]*.

- `state` can now be used with attributes. It used to mean the same thing as `my` if attributes were present *[perl #68658]*.

- Expressions like `@$a > 3` no longer cause `$a` to be mentioned in the "Use of uninitialized value in numeric gt" warning when `$a` is undefined (since it is not part of the > expression, but the operand of the `@`) *[perl #72090]*.

- `require` no longer causes `caller` to return the wrong file name for the scope that called `require` and other scopes higher up that had the same file name *[perl #68712]*.
- The ref types in the typemap for XS bindings now support magical variables *[perl #72684]*.
- Match variables (e.g., `$1`) no longer persist between calls to a sort subroutine *[perl #76026]*.
- The `B` module was returning `B::OP`s instead of `B::LOGOP`s for `entertry` *[perl #80622]*. This was due to a bug in the perl core, not in `B` itself.
- Some numeric operators were converting integers to floating point, resulting in loss of precision on 64-bit platforms *[perl #77456]*.
- The fallback behaviour of overloading on binary operators was asymmetric *[perl #71286]*.

Acknowledgements

Perl 5.13.8 represents approximately one month of development since Perl 5.13.7 and contains 38715 lines of changes across 546 files from 38 authors and committers.

Thank you to the following for contributing to this release:

Abhijit Menon-Sen, Abigail, Andreas König, Ben Morrow, Brad Gilbert, brian d foy, Chip Salzenberg, Chris 'BinGOs' Williams, Craig A. Berry, David Golden, David Leadbeater, David Mitchell, Father Chrysostomos, Florian Ragwitz, Goro Fuji, H.Merijn Brand, Jan Dubois, Jerry D. Hedden, Jesse Vincent, John Peacock, Karl Williamson, Lukas Mai, Marvin Humphrey, Max Maischein, Michael Breen, Michael Fig, Nicholas Clark, Nick Cleaton, Paul Evans, Peter J. Holzer, Peter John Acklam, Rafael Garcia-Suarez, Reini Urban, Renee Baecker, Ricardo Signes, Tony Cook, Yves Orton, Zefram

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.