

## NAME

perl5112delta - what is new for perl v5.11.2

## DESCRIPTION

This document describes differences between the 5.11.1 release and the 5.11.2 release.

## Core Enhancements

### qr overloading

It is now possible to overload the `qr//` operator, that is, conversion to regexp, like it was already possible to overload conversion to boolean, string or number of objects. It is invoked when an object appears on the right hand side of the `=~` operator, or when it is interpolated into a regexp. See *overload*.

### Pluggable keywords

Extension modules can now cleanly hook into the Perl parser to define new kinds of keyword-headed expression and compound statement. The syntax following the keyword is defined entirely by the extension. This allow a completely non-Perl sublanguage to be parsed inline, with the right ops cleanly generated. This feature is currently considered experimental.

See *"PL\_keyword\_plugin" in perlapi* for the mechanism. The Perl core source distribution also includes a new module `XS::APITest::KeywordRPN`, which implements reverse Polish notation arithmetic via pluggable keywords. This module is mainly used for test purposes, and is not normally installed, but also serves as an example of how to use the new mechanism.

### APIs for more internals

The lowest layers of the lexer and parts of the pad system now have C APIs available to XS extensions. These are necessary to support proper use of pluggable keywords, but have other uses too. The new APIs are experimental, and only cover a small proportion of what would be necessary to take full advantage of the core's facilities in these areas. It is intended that the Perl 5.13 development cycle will see the addition of a full range of clean, supported interfaces.

### Overridable function lookup

Where an extension module hooks the creation of `rv2cv` ops to modify the subroutine lookup process, this now works correctly for bareword subroutine calls. This means that prototypes on subroutines referenced this way will be processed correctly. (Previously bareword subroutine names were initially looked up, for parsing purposes, by an unhookable mechanism, so extensions could only properly influence subroutine names that appeared with an `&` sigil.)

## Modules and Pragmata

### New Modules and Pragmata

`legacy`

Preserves legacy behaviors or enable new non-default behaviors. Currently the only behaviour concerns semantics for the 128 characters on ASCII systems that have the 8th bit set.

### Pragmata Changes

`diagnostics`

Supports `%.0f` formatting internally.

`overload`

Allow overloading of `'qr'`.

### Updated Modules

`B::Concise`

---

Optimize reversing an array in-place, avoid using defined %hash in core code and tests.

B::Deparse

Teach B::Deparse about in-place reverse.

Carp

Refine Carp caller() fix and add tests.

Compress::Zlib

Updated to 2.022.

CPANPLUS

Updated to 0.89\_09.

Encode

Updated to 2.38.

ExtUtils::CBuilder

Updated to 0.27.

Env

Add EXISTS and DELETE methods to Env.pm.

File::Fetch

Updated to 0.22.

I18N::Langinfo

Correctly document export of I18N::Langinfo.

I18N::LangTags

In I18N::LangTags::Detect, avoid using defined @array and defined %hash.

IO::Compress

Updated to 2.022.

IPC::Cmd

Updated to 0.54.

List::Util

Updated to 1.22.

Locale::Maketext

In Locale::Maketext, avoid using defined @array and defined %hash. Convert the odd Locale::Maketext test out from Test to Test::More.

Module::Build

Updated to 0.35\_08.

Module::CoreList

Implemented is\_deprecated().

Pod::Simple

Updated to 3.10.

Scalar::Util

Updated to 1.22.

Switch

Updated to 2.16.

## Utility Changes

*a2p*

Fixed bugs with the `match()` operator in list context, remove mention of `$[`.

## Performance Enhancements

- Reversing an array to itself (as in `@a = reverse @a`) in void context now happens in-place and is several orders of magnitude faster than it used to be. It will also preserve non-existent elements whenever possible, i.e. for non magical arrays or tied arrays with `EXISTS` and `DELETE` methods.

## New or Changed Diagnostics

Several new diagnostics, see *perldiag* for details.

Bad plugin affecting keyword '`%s`'

`gmtime(%.0f)` too large

Lexing code attempted to stuff non-Latin-1 character into Latin-1 input

Lexing code internal error (`%s`)

`localtime(%.0f)` too large

Overloaded dereference did not return a reference

Overloaded `qr` did not return a `REGEXP`

`Perl_pmflag()` is deprecated, and will be removed from the XS API

One diagnostic has been removed:

Runaway format

## Changed Internals

- `Perl_pmflag` has been removed from the public API. Calling it now generates a deprecation warning, and it will be removed in a future release. Although listed as part of the API, it was never documented, and only ever used in *toke.c*, and prior to 5.10, *regcomp.c*. In core, it has been replaced by a static function.

## New Tests

*t/op/while\_readdir.t*

Test that a bare `readdir` in while loop sets `$_`.

## Known Problems

Known test failures on VMS

Perl 5.11.2 fails a small set of core and CPAN tests as of this release. With luck, that'll be sorted out for 5.11.3.

## Deprecations

The following items are now deprecated.

### Use of `:=` to mean an empty attribute list is now deprecated.

An accident of Perl's parser meant that these constructions were all equivalent:

```
my $pi := 4;  
my $pi : = 4;  
my $pi : = 4;
```

with the `:` being treated as the start of an attribute list, which ends before the `=`. As whitespace is not significant here, all are parsed as an empty attribute list, hence all the above are equivalent to, and better written as

```
my $pi = 4;
```

because no attribute processing is done for an empty list.

As is, this meant that `:=` cannot be used as a new token, without silently changing the meaning of existing code. Hence that particular form is now deprecated, and will become a syntax error. If it is absolutely necessary to have empty attribute lists (for example, because of a code generator) then avoid the warning by adding a space before the `=`.

## Acknowledgements

Perl 5.11.2 represents approximately 3 weeks development since Perl 5.11.1 and contains 29,992 lines of changes across 458 files from 38 authors and committers:

Abhijit Menon-Sen, Abigail, Ben Morrow, Bo Borgerson, Brad Gilbert, Bram, Chris Williams, Craig A. Berry, Daniel Frederick Crisman, Dave Rolsky, David E. Wheeler, David Golden, Eric Brine, Father Chrysostomos, Frank Wiegand, Gerard Goossen, Gisle Aas, Graham Barr, Harmen, H.Merijn Brand, Jan Dubois, Jerry D. Hedden, Jesse Vincent, Karl Williamson, Kevin Ryde, Leon Brocard, Nicholas Clark, Paul Marquess, Philippe Bruhat, Rafael Garcia-Suarez, Sisyphus, Steffen Mueller, Steve Hay, Steve Peters, Vincent Pit, Yuval Kogman, Yves Orton, and Zefram.

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.