

NAME

perlexperiment - A listing of experimental features in Perl

DESCRIPTION

This document lists the current and past experimental features in the perl core. Although all of these are documented with their appropriate topics, this succinct listing gives you an overview and basic facts about their status.

So far we've merely tried to find and list the experimental features and infer their inception, versions, etc. There's a lot of speculation here.

Current experiments

`our` can now have an experimental optional attribute `unique`

Introduced in Perl 5.8.0

Deprecated in Perl 5.10.0

The ticket for this feature is [\[perl #119313\]](#).

Smart match (`~~`)

Introduced in Perl 5.10.0

Modified in Perl 5.10.1, 5.12.0

Using this feature triggers warnings in the category `experimental::smartmatch`.

The ticket for this feature is [\[perl #119317\]](#).

Pluggable keywords

The ticket for this feature is [\[perl #119455\]](#).

See "[PL_keyword_plugin](#)" in *perlapi* for the mechanism.

Introduced in Perl 5.11.2

Regular Expression Set Operations

Introduced in Perl 5.18

The ticket for this feature is [\[perl #119451\]](#).

See also: "[Extended Bracketed Character Classes](#)" in *perlrecharclass*

Using this feature triggers warnings in the category `experimental::regex_sets`.

Subroutine signatures

Introduced in Perl 5.20.0

Using this feature triggers warnings in the category `experimental::signatures`.

The ticket for this feature is [\[perl #121481\]](#).

Aliasing via reference

Introduced in Perl 5.22.0

Using this feature triggers warnings in the category `experimental::refaliasing`.

The ticket for this feature is [\[perl #122947\]](#).

See also: "[Assigning to References](#)" in *perlref*

The `"const"` attribute

Introduced in Perl 5.22.0

Using this feature triggers warnings in the category `experimental::const_attr`.

The ticket for this feature is [\[perl #123630\]](#).

See also: "[Constant Functions](#)" in *perlsub*

use re 'strict';

Introduced in Perl 5.22.0

Using this feature triggers warnings in the category `experimental::re_strict`.

See *"strict' mode" in re*

String- and number-specific bitwise operators

Introduced in Perl 5.22.0

See also: *"Bitwise String Operators" in perlop*

Using this feature triggers warnings in the category `experimental::bitwise`.

The ticket for this feature is *[perl #123707]*.

The `<:win32>` IO pseudolayer

The ticket for this feature is *[perl #119453]*.

See also *perlrun*

Declaring a reference to a variable

Introduced in Perl 5.26.0

Using this feature triggers warnings in the category `experimental::declared_refs`.

The ticket for this feature is *[perl #128654]*.

See also: *"Declaring a Reference to a Variable" in perlref*

There is an `installhtml` target in the Makefile.

The ticket for this feature is *[perl #116487]*.

Unicode in Perl on EBCDIC

Accepted features

These features were so wildly successful and played so well with others that we decided to remove their experimental status and admit them as full, stable features in the world of Perl, lavishing all the benefits and luxuries thereof. They are also awarded +5 Stability and +3 Charisma.

64-bit support

Introduced in Perl 5.005

`die` accepts a reference

Introduced in Perl 5.005

DB module

Introduced in Perl 5.6.0

See also *perldebug*, *perldebtut*

Weak references

Introduced in Perl 5.6.0

Internal file glob

Introduced in Perl 5.6.0

`fork()` emulation

Introduced in Perl 5.6.1

See also *perlfork*

`-Dusemultiplicity` `-Duseithreads`

Introduced in Perl 5.6.0

Accepted in Perl 5.8.0

Support for long doubles

Introduced in Perl 5.6.0

Accepted in Perl 5.8.1

The `\N` regex character class

The `\N` character class, not to be confused with the named character sequence `\N{NAME}`, denotes any non-newline character in a regular expression.

Introduced in Perl 5.12

Exact version of acceptance unclear, but no later than Perl 5.18.

`(?{code})` and `(??{ code })`

Introduced in Perl 5.6.0

Accepted in Perl 5.20.0

See also *perlre*

Linux abstract Unix domain sockets

Introduced in Perl 5.9.2

Accepted before Perl 5.20.0. The `Socket` library is now primarily maintained on CPAN, rather than in the perl core.

See also *Socket*

Lvalue subroutines

Introduced in Perl 5.6.0

Accepted in Perl 5.20.0

See also *perlsub*

Backtracking control verbs

`(*ACCEPT)`

Introduced in Perl 5.10

Accepted in Perl 5.20.0

The `<:pop>` IO pseudolayer

See also *perlrun*

Accepted in Perl 5.20.0

`\s` in regexp matches vertical tab

Accepted in Perl 5.22.0

Postfix dereference syntax

Introduced in Perl 5.20.0

Accepted in Perl 5.24.0

Lexical subroutines

Introduced in Perl 5.18.0

Accepted in Perl 5.26.0

Removed features

These features are no longer considered experimental and their functionality has disappeared. It's your own fault if you wrote production programs using these features after we explicitly told you not to (see *perlpolicy*).

5.005-style threading

Introduced in Perl 5.005

Removed in Perl 5.10

perlcc

Introduced in Perl 5.005

Moved from Perl 5.9.0 to CPAN

The pseudo-hash data type

Introduced in Perl 5.6.0

Removed in Perl 5.9.0

GetOpt::Long Options can now take multiple values at once (experimental)

`Getopt::Long` upgraded to version 2.35

Removed in Perl 5.8.8

Assertions

The `-A` command line switch

Introduced in Perl 5.9.0

Removed in Perl 5.9.5

Test::Harness::Straps

Moved from Perl 5.10.1 to CPAN

legacy

The experimental `legacy` pragma was swallowed by the `feature` pragma.

Introduced in Perl 5.11.2

Removed in Perl 5.11.3

Lexical `$_`

Using this feature triggered warnings in the category `experimental::lexical_topic`.

Introduced in Perl 5.10.0

Removed in Perl 5.24.0

Array and hash container functions accept references

Using this feature triggered warnings in the category `experimental::autoderef`.

Superseded by *Postfix dereference syntax*.

Introduced in Perl 5.14.0

Removed in Perl 5.24.0

SEE ALSO

For a complete list of features check *feature*.

AUTHORS

brian d foy <brian.d.foy@gmail.com>

Sébastien Aperghis-Tramoni <saper@cpan.org>

COPYRIGHT

Copyright 2010, brian d foy <brian.d.foy@gmail.com>

LICENSE

You can use and redistribute this document under the same terms as Perl itself.