NAME

perl594delta - what is new for perl v5.9.4

DESCRIPTION

This document describes differences between the 5.9.3 and the 5.9.4 development releases. See *perl590delta*, *perl591delta*, *perl592delta* and *perl593delta* for the differences between 5.8.0 and 5.9.3.

Incompatible Changes

chdir FOO

A bareword argument to chdir() is now recognized as a file handle. Earlier releases interpreted the bareword as a directory name. (Gisle Aas)

Handling of pmc files

An old feature of perl was that before `require` or `use` look for a file with a *.pm* extension, they will first look for a similar filename with a *.pmc* extension. If this file is found, it will be loaded in place of any potentially existing file ending in a *.pm* extension.

Previously, *.pmc* files were loaded only if more recent than the matching *.pm* file. Starting with 5.9.4, they'll be always loaded if they exist. (This trick is used by Pugs.)

@- and @+ in patterns

The special arrays `@-` and `@+` are no longer interpolated in regular expressions. (Sadahiro Tomoyuki)

$AUTOLOAD can now be tainted

If you call a subroutine by a tainted name, and if it defers to an AUTOLOAD function, then $AUTOLOAD will be (correctly) tainted. (Rick Delaney)

Core Enhancements

state() variables

A new class of variables has been introduced. State variables are similar to `my` variables, but are declared with the `state` keyword in place of `my`. They're visible only in their lexical scope, but their value is persistent: unlike `my` variables, they're not undefined at scope entry, but retain their previous value. (Rafael Garcia-Suarez)

To use state variables, one needs to enable them by using

```
use feature "state";
```

or by using the `-E` command-line switch in one-liners.

See *"Persistent variables via state()" in perlsub*.

UNIVERSAL::DOES()

The `UNIVERSAL` class has a new method, `DOES()`. It has been added to solve semantic problems with the `isa()` method. `isa()` checks for inheritance, while `DOES()` has been designed to be overridden when module authors use other types of relations between classes (in addition to inheritance). (chromatic)

See *"$obj->DOES(ROLE)" in UNIVERSAL*.

Exceptions in constant folding

The constant folding routine is now wrapped in an exception handler, and if folding throws an exception (such as attempting to evaluate 0/0), perl now retains the current optree, rather than aborting the whole program. (Nicholas Clark, Dave Mitchell)

Source filters in @INC

It's possible to enhance the mechanism of subroutine hooks in @INC by adding a source filter on top of the filehandle opened and returned by the hook. This feature was planned a long time ago, but wasn't quite working until now. See *"require" in perlfunc* for details. (Nicholas Clark)

MAD

MAD, which stands for *Misc Attribute Decoration*, is a still-in-development work leading to a Perl 5 to Perl 6 converter. To enable it, it's necessary to pass the argument `-Dmad` to Configure. The obtained perl isn't binary compatible with a regular perl 5.9.4, and has space and speed penalties; moreover not all regression tests still pass with it. (Larry Wall, Nicholas Clark)

Modules and Pragmas

- `encoding::warnings` is now a lexical pragma. (Although on older perls, which don't have support for lexical pragmas, it keeps its global behaviour.) (Audrey Tang)

- `threads` is now a dual-life module, also available on CPAN. It has been expanded in many ways. A kill() method is available for thread signalling. One can get thread status, or the list of running or joinable threads.

  A new `threads->exit()` method is used to exit from the application (this is the default for the main thread) or from the current thread only (this is the default for all other threads). On the other hand, the exit() built-in now always causes the whole application to terminate. (Jerry D. Hedden)

New Core Modules

- `Hash::Util::FieldHash`, by Anno Siegel, has been added. This module provides support for *field hashes*: hashes that maintain an association of a reference with a value, in a thread-safe garbage-collected way. Such hashes are useful to implement inside-out objects.

- `Module::Build`, by Ken Williams, has been added. It's an alternative to `ExtUtils::MakeMaker` to build and install perl modules.

- `Module::Load`, by Jos Boumans, has been added. It provides a single interface to load Perl modules and *.pl* files.

- `Module::Loaded`, by Jos Boumans, has been added. It's used to mark modules as loaded or unloaded.

- `Package::Constants`, by Jos Boumans, has been added. It's a simple helper to list all constants declared in a given package.

- `Win32API::File`, by Tye McQueen, has been added (for Windows builds). This module provides low-level access to Win32 system API calls for files/dirs.

Utility Changes

config_data

`config_data` is a new utility that comes with `Module::Build`. It provides a command-line interface to the configuration of Perl modules that use Module::Build's framework of configurability (that is, `*::ConfigData` modules that contain local configuration information for their parent modules.)

Documentation

New manpage, perlpragma

The *perlpragma* manpage documents how to write one's own lexical pragmas in pure Perl (something that is possible starting with 5.9.4).

New manpage, perlreguts

The *perlreguts* manpage, courtesy of Yves Orton, describes internals of the Perl regular expression engine.

New manpage, perlunitut

The *perlunitut* manpage is an tutorial for programming with Unicode and string encodings in Perl, courtesy of Juerd Waalboer.

Performance Enhancements

Memory optimisations

Several internal data structures (typeglobs, GVs, CVs, formats) have been restructured to use less memory. (Nicholas Clark)

UTF-8 cache optimisation

The UTF-8 caching code is now more efficient, and used more often. (Nicholas Clark)

Regular expressions

Engine de-recursivised

The regular expression engine is no longer recursive, meaning that patterns that used to overflow the stack will either die with useful explanations, or run to completion, which, since they were able to blow the stack before, will likely take a very long time to happen. If you were experiencing the occasional stack overflow (or segfault) and upgrade to discover that now perl apparently hangs instead, look for a degenerate regex. (Dave Mitchell)

Single char char-classes treated as literals

Classes of a single character are now treated the same as if the character had been used as a literal, meaning that code that uses char-classes as an escaping mechanism will see a speedup. (Yves Orton)

Trie optimisation of literal string alternations

Alternations, where possible, are optimised into more efficient matching structures. String literal alternations are merged into a trie and are matched simultaneously. This means that instead of O(N) time for matching N alternations at a given point the new code performs in O(1) time. (Yves Orton)

Note: Much code exists that works around perl's historic poor performance on alternations. Often the tricks used to do so will disable the new optimisations. Hopefully the utility modules used for this purpose will be educated about these new optimisations by the time 5.10 is released.

Aho-Corasick start-point optimisation

When a pattern starts with a trie-able alternation and there aren't better optimisations available the regex engine will use Aho-Corasick matching to find the start point. (Yves Orton)

Sloppy stat on Windows

On Windows, perl's stat() function normally opens the file to determine the link count and update attributes that may have been changed through hard links. Setting ${^WIN32_SLOPPY_STAT} to a true value speeds up stat() by not performing this operation. (Jan Dubois)

Installation and Configuration Improvements

Relocatable installations

There is now Configure support for creating a relocatable perl tree. If you Configure with `-Duserelocatableinc`, then the paths in @INC (and everything else in %Config) can be optionally located via the path of the perl executable.

That means that, if the string `"..."` is found at the start of any path, it's substituted with the directory of $^X. So, the relocation can be configured on a per-directory basis, although the default with `-Duserelocatableinc` is that everything is relocated. The initial install is done to the original configured prefix.

Ports

Many improvements have been made towards making Perl work correctly on z/OS.

Perl has been reported to work on DragonFlyBSD.

Compilation improvements

All *ppport.h* files in the XS modules bundled with perl are now autogenerated at build time. (Marcus Holland-Moritz)

New probes

The configuration process now detects whether strlcat() and strlcpy() are available. When they are not available, perl's own version is used (from Russ Allbery's public domain implementation). Various places in the perl interpreter now use them. (Steve Peters)

Windows build improvements

Building XS extensions

Support for building XS extension modules with the free MinGW compiler has been improved in the case where perl itself was built with the Microsoft VC++ compiler. (ActiveState)

Support for 64-bit compiler

Support for building perl with Microsoft's 64-bit compiler has been improved. (ActiveState)

Selected Bug Fixes

PERL5SHELL and tainting

On Windows, the PERL5SHELL environment variable is now checked for taintedness. (Rafael Garcia-Suarez)

Using *FILE{IO}

`stat()` and `-X` filetests now treat *FILE{IO} filehandles like *FILE filehandles. (Steve Peters)

Overloading and reblessing

Overloading now works when references are reblessed into another class. Internally, this has been implemented by moving the flag for "overloading" from the reference to the referent, which logically is where it should always have been. (Nicholas Clark)

Overloading and UTF-8

A few bugs related to UTF-8 handling with objects that have stringification overloaded have been fixed. (Nicholas Clark)

eval memory leaks fixed

Traditionally, `eval 'syntax error'` has leaked badly. Many (but not all) of these leaks have now been eliminated or reduced. (Dave Mitchell)

Random device on Windows

In previous versions, perl would read the file */dev/urandom* if it existed when seeding its random number generator. That file is unlikely to exist on Windows, and if it did would probably not contain appropriate data, so perl no longer tries to read it on Windows. (Alex Davies)

New or Changed Diagnostics

State variable %s will be reinitialized

One can assign initial values to state variables, but not when they're declared as a sub-part of a list assignment. See *perldiag*.

Changed Internals

A new file, *mathoms.c*, contains functions that aren't used anymore in the perl core, but remain around because modules out there might still use them. They come from a factorization effort: for example, many PP functions are now shared for several ops.

The implementation of the special variables $^H and %^H has changed, to allow implementing lexical pragmas in pure perl.

Known Problems

One warning test (number 263 in *lib/warnings.t*) fails under UTF-8 locales.

Bytecode tests fail under several platforms. We are considering removing support for byteloader and compiler before the 5.10.0 release.

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/rt3/ . 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.

SEE ALSO

The *Changes* file for 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.