NAME

perl5139delta - what is new for perl v5.13.9

DESCRIPTION

This document describes differences between the 5.13.8 release and the 5.13.9 release.

If you are upgrading from an earlier release such as 5.13.7, first read *perl5138delta*, which describes differences between 5.13.7 and 5.13.8.

Core Enhancements

New regular expression modifier /a

The `/a` regular expression modifier restricts `\s` to match precisely the five characters `[ \f\n\r\t]`, `\d` to match precisely the 10 characters `[0-9]`, `\w` to match precisely the 63 characters `[A-Za-z0-9_]`, and the Posix (`[[:posix:]]`) character classes to match only the appropriate ASCII characters. The complements, of course, match everything but; and `\b` and `\B` are correspondingly affected. Otherwise, `/a` behaves like the `/u` modifier, in that case-insensitive matching uses Unicode semantics; for example, "k" will match the Unicode `\N{KELVIN SIGN}` under `/i` matching, and code points in the Latin1 range, above ASCII will have Unicode semantics when it comes to case-insensitive matching. Like its cousins (`/u`, `/l`, and `/d`), and in spite of the terminology, `/a` in 5.14 will not actually be able to be used as a suffix at the end of a regular expression (this restriction is planned to be lifted in 5.16). It must occur either as an infix modifier, such as `(?a:...)` or `((?a)...`, or it can be turned on within the lexical scope of `use re '/a'`. Turning on `/a` turns off the other "character set" modifiers.

Any unsigned value can be encoded as a character

With this release, Perl is adopting a model that any unsigned value can be treated as a code point and encoded internally (as utf8) without warnings -- not just the code points that are legal in Unicode. However, unless utf8 warnings have been explicitly lexically turned off, outputting or performing a Unicode-defined operation (such as upper-casing) on such a code point will generate a warning. Attempting to input these using strict rules (such as with the `:encoding('UTF-8')` layer) will continue to fail. Prior to this release the handling was very inconsistent, and incorrect in places. Also, the Unicode non-characters, some of which previously were erroneously considered illegal in places by Perl, contrary to the Unicode standard, are now always legal internally. But inputting or outputting them will work the same as for the non-legal Unicode code points, as the Unicode standard says they are illegal for "open interchange".

Regular expression debugging output improvement

Regular expression debugging output (turned on by `use re 'debug';`) now uses hexadecimal when escaping non-ASCII characters, instead of octal.

Security

Restrict \p{IsUserDefined} to In\w+ and Is\w+

In *"User-Defined Character Properties" in perlunicode*, it says you can create custom properties by defining subroutines whose names begin with "In" or "Is". However, perl doesn't actually enforce that naming restriction, so \p{foo::bar} will call foo::Bar() if it exists.

This commit finally enforces this convention. Note that this broke a number of existing tests for properties, since they didn't always use an Is/In prefix.

Incompatible Changes

All objects are destroyed

It used to be possible to prevent a destructor from being called during global destruction by artificially increasing the reference count of an object.

Now such objects *will* will be destroyed, as a result of a bug fix *[perl #81230]*.

This has the potential to break some XS modules. (In fact, it break some. See *Known Problems*, below.)

Modules and Pragmata

New Modules and Pragmata

- `CPAN::Meta::YAML` 0.003 has been added as a dual-life module. It supports a subset of YAML sufficient for reading and writing META.yml and MYMETA.yml files included with CPAN distributions or generated by the module installation toolchain. It should not be used for any other general YAML parsing or generation task.

- `HTTP::Tiny` 0.009 has been added as a dual-life module. It is a very small, simple HTTP/1.1 client designed for simple GET requests and file mirroring. It has has been added to enable CPAN.pm and CPANPLUS to "bootstrap" HTTP access to CPAN using pure Perl without relying on external binaries like *curl* or *wget*.

- `JSON::PP` 2.27103 has been added as a dual-life module, for the sake of reading *META.json* files in CPAN distributions.

- `Module::Metadata` 1.000003 has been added as a dual-life module. It gathers package and POD information from Perl module files. It is a standalone module based on Module::Build::ModuleInfo for use by other module installation toolchain components. Module::Build::ModuleInfo has been deprecated in favor of this module instead.

- `Perl::OSType` 1.002 has been added as a dual-life module. It maps Perl operating system names (e.g. 'dragonfly' or 'MSWin32') to more generic types with standardized names (e.g. "Unix" or "Windows"). It has been refactored out of Module::Build and ExtUtils::CBuilder and consolidates such mappings into a single location for easier maintenance.

Updated Modules and Pragmata

- `Archive::Extract` has been upgraded from version 0.46 to 0.48

- `Archive::Tar` has been upgraded from version 1.74 to 1.76

- `CGI` has been upgraded from version 3.50 to 3.51

  Further improvements have been made to guard against newline injections in headers.

- `Compress::Raw::Bzip2` has been upgraded from version 2.031 to 2.033

- `Compress::Raw::Zlib` has been upgraded from version 2.030 to 2.033

- `CPAN` has been upgraded from version 1.94_62 to 1.94_63

- `CPANPLUS` has been upgraded from version 0.9010 to 0.9011

- `CPANPLUS::Dist::Build` has been upgraded from version 0.50 to 0.52

- `DB_File` has been upgraded from version 1.820 to 1.821

- `Encode` has been upgraded from version 2.40 to 2.42. Now, all 66 Unicode non-characters are treated the same way U+FFFF has always been treated; if it was disallowed, all 66 are disallowed; if it warned, all 66 warn.

- `File::Fetch` has been upgraded from version 0.28 to 0.32

- `IO::Compress` has been upgraded from version 2.030 to 2.033

- `IPC::Cmd` has been upgraded from version 0.66 to 0.68

- `Log::Message` has been upgraded from version 0.02 to 0.04

- `Log::Message::Simple` has been upgraded from version 0.06 to 0.08

- `Module::Load::Conditional` has been upgraded from version 0.38 to 0.40

- `Object::Accessor` has been upgraded from version 0.36 to 0.38

- `Params::Check` has been upgraded from version 0.26 to 0.28

- `Pod::LaTeX` has been upgraded from version 0.58 to 0.59

- `Socket` has been updated with new affordances for IPv6, including implementations of the `Socket::getaddrinfo()` and `Socket::getnameinfo()` functions, along with related constants.

- `Term::UI` has been upgraded from version 0.20 to 0.24

- `Thread::Queue` has been upgraded from version 2.11 to 2.12.

- `Thread::Semaphore` has been upgraded from version 2.11 to 2.12.

- `threads` has been upgraded from version 1.81_03 to 1.82

- `threads::shared` has been upgraded from version 1.35 to 1.36

- `Time::Local` has been upgraded from version 1.1901_01 to 1.2000.

- `Unicode::Normalize` has been upgraded from version 1.07 to 1.10

- `version` has been upgraded from 0.86 to 0.88.

- `Win32` has been upgraded from version 0.41 to 0.44.

Documentation

Changes to Existing Documentation

All documentation

- Numerous POD warnings were fixed.

- Many, many spelling errors and typographical mistakes were corrected throughout Perl's core.

perlhack

- `perlhack` was extensively reorganized.

perlfunc

- It has now been documented that `ord` returns 0 for an empty string.

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

- Performing an operation requiring Unicode semantics (such as case-folding) on a Unicode surrogate or a non-Unicode character now triggers a warning: 'Operation "%s" returns its argument for ...'.

Changes to Existing Diagnostics

- Previously, if none of the `gethostbyaddr`, `gethostbyname` and `gethostent` functions were implemented on a given platform, they would all die with the message 'Unsupported socket function "gethostent" called', with analogous messages for `getnet*` and `getserv*`. This has been corrected.

Utility Changes

perlbug

- `perlbug` did not previously generate a From: header, potentially resulting in dropped mail. Now it does include that header.

buildtoc

- *pod/buildtoc* has been modernized and can now be used to test the well-formedness of *pod/perltoc.pod* automatically.

Testing

- `lib/File/DosGlob.t` has been modernized and now uses `Test::More`.

- A new test script, `t/porting/filenames.t`, makes sure that filenames and paths are reasonably portable.

- `t/porting/diag.t` is now several orders of magnitude faster.

- `t/porting/buildtoc.t` now tests that the documentation TOC file is current and well-formed.

- `t/base/while.t` now tests the basics of a while loop with minimal dependencies.

- `t/cmd/while.t` now uses *test.pl* for better maintainability.

- `t/op/split.t` now tests calls to `split` without any pattern specified.

Platform Support

Discontinued Platforms

Apollo DomainOS

The last vestiges of support for this platform have been excised from the Perl distribution. It was officially discontinued in version 5.12.0. It had not worked for years before that.

MacOS Classic

The last vestiges of support for this platform have been excised from the Perl distribution. It was officially discontinued in an earlier version.

Platform-Specific Notes

Cygwin

- Updated MakeMaker to build man pages on cygwin.

- Improved rebase behaviour

    If a dll is updated on cygwin reuse the old imagebase address. This solves most rebase errors, esp when updating on core dll's. See *http://www.tishler.net/jason/software/rebase/rebase-2.4.2.README* for more information.

- Support the standard cyg dll prefix, which is e.g. needed for FFI's.

- Updated build hints file

Solaris

DTrace is now supported on Solaris. There used to be build failures, but these have been fixed *[perl #73630]*.

Internal Changes

- The opcode bodies for `chop` and `chomp` and for `schop` and `schomp` have been merged. The implementation functions `Perl_do_chop()` and `Perl_do_chomp()`, never part of the public

API, have been merged and moved to a static function in *pp.c*. This shrinks the perl binary slightly, and should not affect any code outside the core (unless it is relying on the order of side effects when `chomp` is passed a *list* of values).

- Some of the flags parameters to the uvuni_to_utf8_flags() and utf8n_to_uvuni() have changed. This is a result of Perl now allowing internal storage and manipulation of code points that are problematic in some situations. Hence, the default actions for these functions has been complemented to allow these code points. The new flags are documented in *perlapi*. Code that requires the problematic code points to be rejected needs to change to use these flags. Some flag names are retained for backward source compatibility, though they do nothing, as they are now the default. However the flags `UNICODE_ALLOW_FDD0`, `UNICODE_ALLOW_FFFF`, `UNICODE_ILLEGAL`, and `UNICODE_IS_ILLEGAL` have been removed, as they stem from a fundamentally broken model of how the Unicode non-character code points should be handled, which is now described in *"Non-character code points" in perlunicode*. See also *Selected Bug Fixes*.

- Certain shared flags in the `pmop.op_pmflags` and `regexp.extflags` structures have been removed. These are: `Rxf_Pmf_LOCALE`, `Rxf_Pmf_UNICODE`, and `PMf_LOCALE`. Instead there are encodes and three static in-line functions for accessing the information: `get_regex_charset()`, `set_regex_charset()`, and `get_regex_charset_name()`, which are defined in the places where the original flags were.

- A new option has been added to `pv_escape` to dump all characters above ASCII in hexadecimal. Before, one could get all characters as hexadecimal or the Latin1 non-ASCII as octal

- Generate pp_* prototypes in pp_proto.h, and remove pp.sym

  Eliminate the #define pp_foo Perl_pp_foo(pTHX) macros, and update the 13 locations that relied on them.

  regen/opcode.pl now generates prototypes for the PP functions directly, into pp_proto.h. It no longer writes pp.sym, and regen/embed.pl no longer reads this, removing the only ordering dependency in the regen scripts. opcode.pl is now responsible for prototypes for pp_* functions. (embed.pl remains responsible for ck_* functions, reading from regen/opcodes)

Selected Bug Fixes

- The handling of Unicode non-characters has changed. Previously they were mostly considered illegal, except that only one of the 66 of them was known about in places. The Unicode standard considers them legal, but forbids the "open interchange" of them. This is part of the change to allow the internal use of any code point (see *Core Enhancements*). Together, these changes resolve *# 38722*, *# 51918*, *# 51936*, *# 63446*

- Sometimes magic (ties, tainted, etc.) attached to variables could cause an object to last longer than it should, or cause a crash if a tied variable were freed from within a tie method. These have been fixed *[perl #81230]*.

- Most I/O functions were not warning for unopened handles unless the 'closed' and 'unopened' warnings categories were both enabled. Now only `use warnings 'unopened'` is necessary to trigger these warnings (as was always meant to be the case.

- `<expr>` always respects overloading now if the expression is overloaded.

  Due to the way that '<> as glob' was parsed differently from '<> as filehandle' from 5.6 onwards, something like `<$foo[0]>` did not handle overloading, even if `$foo[0]` was an overloaded object. This was contrary to the documentation for overload, and meant that `<>` could not be used as a general overloaded iterator operator.

- Destructors on objects were not called during global destruction on objects that were not referenced by any scalars. This could happen if an array element were blessed (e.g., `bless \$a[0]`) or if a closure referenced a blessed variable (`bless \my @a; sub foo { @a }`).

Now there is an extra pass during global destruction to fire destructors on any objects that might be left after the usual passes that check for objects referenced by scalars *[perl #36347]*.

- A long standing bug has now been fully fixed (partial fixes came in earlier releases), in which some Latin-1 non-ASCII characters on ASCII-platforms would match both a character class and its complement, such as U+00E2 being both in `\w` and `\W`, depending on the UTF-8-ness of the regular expression pattern and target string. Fixing this did expose some bugs in various modules and tests that relied on the previous behavior of `[[:alpha:]]` not ever matching U+00FF, "LATIN SMALL LETTER Y WITH DIAERESIS", even when it should, in Unicode mode; now it does match when appropriate. *[perl #60156]*.

Known Problems

- The fix for [perl #81230] causes test failures for `Tk` version 804.029. This is still being investigated.

Acknowledgements

Perl 5.13.9 represents approximately one month of development since Perl 5.13.8 and contains approximately 48000 lines of changes across 809 files from 35 authors and committers:

Abigail, Ãvar ArnfjÃ¶rÃ° Bjarmason, brian d foy, Chris 'BinGOs' Williams, Craig A. Berry, David Golden, David Leadbeater, David Mitchell, Father Chrysostomos, Florian Ragwitz, Gerard Goossen, H.Merijn Brand, Jan Dubois, Jerry D. Hedden, Jesse Vincent, John Peacock, Karl Williamson, Leon Timmermans, Michael Parker, Michael Stevens, Nicholas Clark, Nuno Carvalho, Paul "LeoNerd" Evans, Peter J. Acklam, Peter Martini, Rainer Tammer, Reini Urban, Renee Baecker, Ricardo Signes, Robin Barker, Tony Cook, Vadim Konovalov, Vincent Pit, Zefram, and ZsbÃ¡n Ambrus.

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.