

## NAME

perl592delta - what is new for perl v5.9.2

## DESCRIPTION

This document describes differences between the 5.9.1 and the 5.9.2 development releases. See *perl590delta* and *perl591delta* for the differences between 5.8.0 and 5.9.1.

## Incompatible Changes

### Packing and UTF-8 strings

The semantics of `pack()` and `unpack()` regarding UTF-8-encoded data has been changed. Processing is now by default character per character instead of byte per byte on the underlying encoding. Notably, code that used things like `pack("a*", $string)` to see through the encoding of string will now simply get back the original `$string`. Packed strings can also get upgraded during processing when you store upgraded characters. You can get the old behaviour by using `use bytes`.

To be consistent with `pack()`, the `C0` in `unpack()` templates indicates that the data is to be processed in character mode, i.e. character by character; on the contrary, `U0` in `unpack()` indicates UTF-8 mode, where the packed string is processed in its UTF-8-encoded Unicode form on a byte by byte basis. This is reversed with regard to perl 5.8.X.

Moreover, `C0` and `U0` can also be used in `pack()` templates to specify respectively character and byte modes.

`C0` and `U0` in the middle of a `pack` or `unpack` format now switch to the specified encoding mode, honoring parens grouping. Previously, parens were ignored.

Also, there is a new `pack()` character format, `w`, which is intended to replace the old `C`. `C` is kept for unsigned chars coded as bytes in the strings internal representation. `w` represents unsigned (logical) character values, which can be greater than 255. It is therefore more robust when dealing with potentially UTF-8-encoded data (as `C` will wrap values outside the range 0..255, and not respect the string encoding).

In practice, that means that `pack` formats are now encoding-neutral, except `C`.

For consistency, `A` in `unpack()` format now trims all Unicode whitespace from the end of the string. Before perl 5.9.2, it used to strip only the classical ASCII space characters.

## Miscellaneous

The internal dump output has been improved, so that non-printable characters such as newline and backspace are output in `\x` notation, rather than octal.

The `-C` option can no longer be used on the `#!` line. It wasn't working there anyway.

## Core Enhancements

### Malloc wrapping

Perl can now be built to detect attempts to assign pathologically large chunks of memory. Previously such assignments would suffer from integer wrap-around during size calculations causing a misallocation, which would crash perl, and could theoretically be used for "stack smashing" attacks. The wrapping defaults to enabled on platforms where we know it works (most AIX configurations, BSDi, Darwin, DEC OSF/1, FreeBSD, HP-UX, GNU Linux, OpenBSD, Solaris, VMS and most Win32 compilers) and defaults to disabled on other platforms.

### Unicode Character Database 4.0.1

The copy of the Unicode Character Database included in Perl 5.9 has been updated to 4.0.1 from 4.0.0.

## suidperl less insecure

Paul Szabo has analysed and patched `suidperl` to remove existing known insecurities. Currently there are no known holes in `suidperl`, but previous experience shows that we cannot be confident that these were the last. You may no longer invoke the set uid perl directly, so to preserve backwards compatibility with scripts that invoke `#!/usr/bin/suidperl` the only set uid binary is now `sperl5.9.n` (`sperl5.9.2` for this release). `suidperl` is installed as a hard link to `perl`; both `suidperl` and `perl` will invoke `sperl5.9.2` automatically the set uid binary, so this change should be completely transparent.

For new projects the core perl team would strongly recommend that you use dedicated, single purpose security tools such as `sudo` in preference to `suidperl`.

## PERLIO\_DEBUG

The `PERLIO_DEBUG` environment variable has no longer any effect for setuid scripts and for scripts run with `-T`.

Moreover, with a thread-enabled perl, using `PERLIO_DEBUG` could lead to an internal buffer overflow. This has been fixed.

## Formats

In addition to bug fixes, `format`'s features have been enhanced. See *perlfm*.

## Unicode Character Classes

Perl's regular expression engine now contains support for matching on the intersection of two Unicode character classes. You can also now refer to user-defined character classes from within other user defined character classes.

## Byte-order modifiers for pack() and unpack()

There are two new byte-order modifiers, `>` (big-endian) and `<` (little-endian), that can be appended to most `pack()` and `unpack()` template characters and groups to force a certain byte-order for that type or group. See *"pack" in perlfunc* and *perlpacktut* for details.

## Byte count feature in pack()

A new `pack()` template character, `" . "`, returns the number of characters read so far.

## New variables

A new variable, `$_{^RE_DEBUG_FLAGS}`, controls what debug flags are in effect for the regular expression engine when running under `use re "debug"`. See *re* for details.

A new variable `$_{^UTF8LOCALE}` indicates where a UTF-8 locale was detected by perl at startup.

## Modules and Pragmata

### New modules

- `encoding::warnings`, by Audrey Tang, is a module to emit warnings whenever an ASCII character string containing high-bit bytes is implicitly converted into UTF-8.
- `Module::CoreList`, by Richard Clamp, is a small handy module that tells you what versions of core modules ship with any versions of Perl 5. It comes with a command-line frontend, `corelist`.

## Updated And Improved Modules and Pragmata

Dual-lived modules have been updated to be kept up-to-date with respect to CPAN.

The dual-lived modules which contain an `_` in their version number are actually *ahead* of the corresponding CPAN release.

B::Concise

B::Concise was significantly improved.

#### Socket

There is experimental support for Linux abstract Unix domain sockets.

#### Sys::Syslog

`syslog()` can now use numeric constants for facility names and priorities, in addition to strings.

#### threads

Detached threads are now also supported on Windows.

## Utility Changes

- The `corelist` utility is now installed with perl (see *New modules* above).
- `h2ph` and `h2xs` have been made a bit more robust with regard to "modern" C code.
- Several bugs have been fixed in `find2perl`, regarding `-exec` and `-eval`. Also the options `-path`, `-ipath` and `-iname` have been added.
- The Perl debugger can now save all debugger commands for sourcing later; notably, it can now emulate stepping backwards, by restarting and rerunning all bar the last command from a saved command history.  
It can also display the parent inheritance tree of a given class.  
Perl has a new `-dt` command-line flag, which enables threads support in the debugger.

## Performance Enhancements

- Unicode case mappings (`/i`, `lc`, `uc`, etc) are faster.
- `@a = sort @a` was optimized to do in-place sort. Likewise, `reverse sort ...` is now optimized to sort in reverse, avoiding the generation of a temporary intermediate list.
- Unnecessary assignments are optimised away in

```
my $s = undef;
my @a = ();
my %h = ();
```
- `map` in scalar context is now optimized.
- The regexp engine now implements the trie optimization : it's able to factorize common prefixes and suffixes in regular expressions. A new special variable, `${^RE_TRIE_MAXBUF}`, has been added to fine-tune this optimization.

## Installation and Configuration Improvements

Run-time customization of `@INC` can be enabled by passing the `-Dusesitecustomize` flag to configure. When enabled, this will make perl run `$sitelibexp/sitecustomize.pl` before anything else. This script can then be set up to add additional entries to `@INC`.

There is alpha support for relocatable `@INC` entries.

Perl should build on Interix and on GNU/kFreeBSD.

## Selected Bug Fixes

Most of those bugs were reported in the perl 5.8.x maintenance track. Notably, quite a few utf8 bugs were fixed, and several memory leaks were suppressed. The perl58Xdelta manpages have more details on them.

Development-only bug fixes include :

`$Foo::_` was wrongly forced as `$main::_`.

## New or Changed Diagnostics

A new warning, `!=~` should be `!~`, is emitted to prevent this misspelling of the non-matching operator.

The warning *Newline in left-justified string* has been removed.

The error *Too late for "-T" option* has been reformulated to be more descriptive.

There is a new compilation error, *Illegal declaration of subroutine*, for an obscure case of syntax errors.

The diagnostic output of Carp has been changed slightly, to add a space after the comma between arguments. This makes it much easier for tools such as web browsers to wrap it, but might confuse any automatic tools which perform detailed parsing of Carp output.

`perl -V` has several improvements, making it more useable from shell scripts to get the value of configuration variables. See *perlrun* for details.

## Changed Internals

The perl core has been refactored and reorganised in several places. In short, this release will not be binary compatible with any previous perl release.

## Known Problems

For threaded builds, *ext/threads/shared/t/wait.t* has been reported to fail some tests on HP-UX 10.20.

*Net::Ping* might fail some tests on HP-UX 11.00 with the latest OS upgrades.

*t/io/dup.t*, *t/io/open.t* and *lib/ExtUtils/t/Constant.t* fail some tests on some BSD flavours.

## Plans for the next release

The current plan for perl 5.9.3 is to add CPANPLUS as a core module. More regular expression optimizations are also in the works.

It is planned to release a development version of perl more frequently, i.e. each time something major changes.

## 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://bugs.perl.org/>. 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.