

NAME

perldelta - what is new for perl v5.8.9

DESCRIPTION

This document describes differences between the 5.8.8 release and the 5.8.9 release.

Notice

The 5.8.9 release will be the last significant release of the 5.8.x series. Any future releases of 5.8.x will likely only be to deal with security issues, and platform build failures. Hence you should look to migrating to 5.10.x, if you have not started already. See *Known Problems* for more information.

Incompatible Changes

A particular construction in the source code of extensions written in C++ may need changing. See *Changed Internals* for more details. All extensions written in C, most written in C++, and all existing compiled extensions are unaffected. This was necessary to improve C++ support.

Other than this, there are no changes intentionally incompatible with 5.8.8. If any exist, they are bugs and reports are welcome.

Core Enhancements**Unicode Character Database 5.1.0.**

The copy of the Unicode Character Database included in Perl 5.8 has been updated to 5.1.0 from 4.1.0. See <http://www.unicode.org/versions/Unicode5.1.0/#NotableChanges> for the notable changes.

stat and -X on directory handles

It is now possible to call `stat` and the `-X` filestat operators on directory handles. As both directory and file handles are barewords, there can be ambiguities over which was intended. In these situations the file handle semantics are preferred. Both also treat `*FILE{IO}` filehandles like `*FILE` filehandles.

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)

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. Without this change, programs would not compile if they had expressions that happened to generate exceptions, even though those expressions were in code that could never be reached at runtime. (Nicholas Clark, Dave Mitchell)

no VERSION

You can now use `no` followed by a version number to specify that you want to use a version of perl older than the specified one.

Improved internal UTF-8 caching code

The code that caches calculated UTF-8 byte offsets for character offsets for a string has been re-written. Several bugs have been located and eliminated, and the code now makes better use of the information it has, so should be faster. In particular, it doesn't scan to the end of a string before calculating an offset within the string, which should speed up some operations on long strings. It is now possible to disable the caching code at run time, to verify that it is not the cause of suspected problems.

Runtime relocatable installations

There is now *Configure* support for creating a perl tree that is relocatable at run time. see *Relocatable installations*.

New internal variables

`${^CHILD_ERROR_NATIVE}`

This variable gives the native status returned by the last pipe close, backtick command, successful call to `wait` or `waitpid`, or from the `system` operator. See *perlvar* for details. (Contributed by Gisle Aas.)

`${^UTF8CACHE}`

This variable controls the state of the internal UTF-8 offset caching code. 1 for on (the default), 0 for off, -1 to debug the caching code by checking all its results against linear scans, and panicing on any discrepancy.

readpipe is now overridable

The built-in function `readpipe` is now overridable. Overriding it permits also to override its operator counterpart, `qx//` (also known as ```).

simple exception handling macros

Perl 5.8.9 (and 5.10.0 onwards) now provides a couple of macros to do very basic exception handling in XS modules. You can use these macros if you call code that may `croak`, but you need to do some cleanup before giving control back to Perl. See *"Exception Handling" in perlguts* for more details.

-D option enhancements

- `-Dq` suppresses the *EXECUTING...* message when running under `-D`
- `-Dl` logs runops loop entry and exit, and jump level popping.
- `-Dv` displays the process id as part of the trace output.

XS-assisted SWASHGET

Some pure-perl code that the regexp engine was using to retrieve Unicode properties and transliteration mappings has been reimplemented in XS for faster execution. (SADAHIRO Tomoyuki)

Constant subroutines

The interpreter internals now support a far more memory efficient form of inlineable constants. Storing a reference to a constant value in a symbol table is equivalent to a full typeglob referencing a constant subroutine, but using about 400 bytes less memory. This proxy constant subroutine is automatically upgraded to a real typeglob with subroutine if necessary. The approach taken is analogous to the existing space optimisation for subroutine stub declarations, which are stored as plain scalars in place of the full typeglob.

However, to aid backwards compatibility of existing code, which (wrongly) does not expect anything other than typeglobs in symbol tables, nothing in core uses this feature, other than the regression tests.

Stubs for prototyped subroutines have been stored in symbol tables as plain strings, and stubs for unprototyped subroutines as the number -1, since 5.005, so code which assumes that the core only places typeglobs in symbol tables has been making incorrect assumptions for over 10 years.

New Platforms

Compile support added for:

- DragonFlyBSD
- MidnightBSD

- MirOS BSD
- RISC OS
- Cray XT4/Catamount

Modules and Pragmata

New Modules

- `Module::Pluggable` is a simple framework to create modules that accept pluggable sub-modules. The bundled version is 3.8
- `Module::CoreList` is a hash of hashes that is keyed on perl version as indicated in `$]`. The bundled version is 2.17
- `Win32API::File` now available in core on Microsoft Windows. The bundled version is 0.1001_01
- `Devel::InnerPackage` finds all the packages defined by a single file. It is part of the `Module::Pluggable` distribution. The bundled version is 0.3

Updated Modules

- `attributes` upgraded to version 0.09
- `AutoLoader` upgraded to version 5.67
- `AutoSplit` upgraded to 1.06
- `autouse` upgraded to version 1.06
- `B` upgraded from 1.09_01 to 1.19
 - provides new pad related abstraction macros `B::NV::COP_SEQ_RANGE_LOW`, `B::NV::COP_SEQ_RANGE_HIGH`, `B::NV::PARENT_PAD_INDEX`, `B::NV::PARENT_FAKELEX_FLAGS`, which hides the difference in storage in 5.10.0 and later.
 - provides `B::sub_generation`, which exposes `PL_sub_generation`
 - provides `B::GV::isGV_with_GP`, which on pre-5.10 perls always returns true.
 - New type `B::HE` added with methods `VAL`, `HASH` and `SVKEY_force`
 - The `B::GVf_IMPORTED_CV` flag is now set correctly when a proxy constant subroutine is imported.
 - bugs fixed in the handling of `PMOPS`.
 - `B::BM::PREVIOUS` returns now `U32`, not `U16`. `B::CV::START` and `B::CV::ROOT` return now `NULL` on an `XSUB`, `B::CV::XSUB` and `B::CV::XSUBANY` return 0 on a non-`XSUB`.
- `B::C` upgraded to 1.05
- `B::Concise` upgraded to 0.76
 - new option `-src` causes the rendering of each statement (starting with the nextstate OP) to be preceded by the first line of source code that generates it.
 - new option `-stash="somepackage"`, requires "somepackage", and then renders each function defined in its namespace.
 - now has documentation of detailed hint symbols.

- B::Debug upgraded to version 1.05
- B::Deparse upgraded to version 0.87
 - properly deparse `print readpipe $x, $y`.
 - now handles `''-(>, ::(), sub :: {}, etc.` correctly [RT #43010]. All bugs in parsing these kinds of syntax are now fixed:

```
perl -MO=Deparse -e '"my %h = "->()'
perl -MO=Deparse -e '::->()'
perl -MO=Deparse -e 'sub :: {}'
perl -MO=Deparse -e 'package a; sub a::b::c {}'
perl -MO=Deparse -e 'sub the::main::road {}'
```
 - does **not** deparse `$$H{v_string}`, which is automatically set by the internals.
- B::Lint upgraded to version 1.11
- B::Terse upgraded to version 1.05
- base upgraded to version 2.13
 - loading a module via `base.pm` would mask a global `$SIG{__DIE__}` in that module.
 - push all classes at once in `@ISA`
- Benchmark upgraded to version 1.10
- bigint upgraded to 0.23
- bignum upgraded to 0.23
- bigrat upgraded to 0.23
- blib upgraded to 0.04
- Carp upgraded to version 1.10

The argument `backtrace` code now shows `undef` as `undef`, instead of a string `"undef"`.
- CGI upgraded to version 3.42
- charnames upgraded to 1.06
- constant upgraded to version 1.17
- CPAN upgraded to version 1.9301
- Cwd upgraded to version 3.29 with some platform specific improvements (including for VMS).
- Data::Dumper upgraded to version 2.121_17
 - Fixes hash iterator current position with the pure Perl version [RT #40668]
 - Performance enhancements, which will be most evident on platforms where repeated calls to C's `realloc()` are slow, such as Win32.
- DB_File upgraded to version 1.817
- DB_Filter upgraded to version 0.02
- Devel::DProf upgraded to version 20080331.00
- Devel::Peek upgraded to version 1.04

- `Devel::PPPort` upgraded to version 3.14
- `diagnostics` upgraded to version 1.16
- `Digest` upgraded to version 1.15
- `Digest::MD5` upgraded to version 2.37
- `DirHandle` upgraded to version 1.02
 - now localises `$.`, `$@`, `$!`, `$^E`, and `$?` before closing the directory handle to suppress leaking any side effects of warnings about it already being closed.
- `DynaLoader` upgraded to version 1.09

`DynaLoader` can now dynamically load a loadable object from a file with a non-default file extension.
- `Encode` upgraded to version 2.26

`Encode::Alias` includes a fix for encoding "646" on Solaris (better known as ASCII).
- `English` upgraded to version 1.03
- `Errno` upgraded to version 1.10
- `Exporter` upgraded to version 5.63
- `ExtUtils::Command` upgraded to version 1.15
- `ExtUtils::Constant` upgraded to version 0.21
- `ExtUtils::Embed` upgraded to version 1.28
- `ExtUtils::Install` upgraded to version 1.50_01
- `ExtUtils::Installed` upgraded to version 1.43
- `ExtUtils::MakeMaker` upgraded to version 6.48
 - support for `INSTALLSITESCRIPT` and `INSTALLVENDORSRIPT` configuration.
- `ExtUtils::Manifest` upgraded to version 1.55
- `ExtUtils::ParseXS` upgraded to version 2.19
- `Fatal` upgraded to version 1.06
 - allows built-ins in `CORE::GLOBAL` to be made fatal.
- `Fcntl` upgraded to version 1.06
- `fields` upgraded to version 2.12
- `File::Basename` upgraded to version 2.77
- `FileCache` upgraded to version 1.07
- `File::Compare` upgraded to 1.1005
- `File::Copy` upgraded to 2.13
 - now uses 3-arg `open`.
- `File::DosGlob` upgraded to 1.01
- `File::Find` upgraded to version 1.13

- `File::Glob` upgraded to version 1.06
 - fixes spurious results with brackets inside braces.
- `File::Path` upgraded to version 2.07_02
- `File::Spec` upgraded to version 3.29
 - improved handling of bad arguments.
 - some platform specific improvements (including for VMS and Cygwin), with an optimisation on `abs2rel` when handling both relative arguments.
- `File::stat` upgraded to version 1.01
- `File::Temp` upgraded to version 0.20
- `filetest` upgraded to version 1.02
- `Filter::Util::Call` upgraded to version 1.07
- `Filter::Simple` upgraded to version 0.83
- `FindBin` upgraded to version 1.49
- `GDBM_File` upgraded to version 1.09
- `Getopt::Long` upgraded to version 2.37
- `Getopt::Std` upgraded to version 1.06
- `Hash::Util` upgraded to version 0.06
- `if` upgraded to version 0.05
- `IO` upgraded to version 1.23
 - Reduced number of calls to `getpeername` in `IO::Socket`
- `IPC::Open` upgraded to version 1.03
- `IPC::Open3` upgraded to version 1.03
- `IPC::SysV` upgraded to version 2.00
- `lib` upgraded to version 0.61
 - avoid warning about loading `.par` files.
- `libnet` upgraded to version 1.22
- `List::Util` upgraded to 1.19
- `Locale::Maketext` upgraded to 1.13
- `Math::BigFloat` upgraded to version 1.60
- `Math::BigInt` upgraded to version 1.89
- `Math::BigRat` upgraded to version 0.22
 - implements new `as_float` method.
- `Math::Complex` upgraded to version 1.54.
- `Math::Trig` upgraded to version 1.18.

- NDBM_File upgraded to version 1.07
 - improve `g++` handling for systems using GDBM compatibility headers.
- Net::Ping upgraded to version 2.35
- NEXT upgraded to version 0.61
 - fix several bugs with NEXT when working with AUTOLOAD, eval block, and within overloaded stringification.
- ODBM_File upgraded to 1.07
- open upgraded to 1.06
- ops upgraded to 1.02
- PerlIO::encoding upgraded to version 0.11
- PerlIO::scalar upgraded to version 0.06
 - [RT #40267] PerlIO::scalar doesn't respect readonly-ness.
- PerlIO::via upgraded to version 0.05
- Pod::Html upgraded to version 1.09
- Pod::Parser upgraded to version 1.35
- Pod::Usage upgraded to version 1.35
- POSIX upgraded to version 1.15
 - POSIX constants that duplicate those in Fcntl are now imported from Fcntl and re-exported, rather than being duplicated by POSIX
 - POSIX::remove can remove empty directories.
 - POSIX::setlocale safer to call multiple times.
 - POSIX::SigRt added, which provides access to POSIX realtime signal functionality on systems that support it.
- re upgraded to version 0.06_01
- Safe upgraded to version 2.16
- Scalar::Util upgraded to 1.19
- SDBM_File upgraded to version 1.06
- SelfLoader upgraded to version 1.17
- Shell upgraded to version 0.72
- sigtrap upgraded to version 1.04
- Socket upgraded to version 1.81
 - this fixes an optimistic use of `gethostbyname`
- Storable upgraded to 2.19
- Switch upgraded to version 2.13
- Sys::Syslog upgraded to version 0.27

- Term::ANSIColor upgraded to version 1.12
- Term::Cap upgraded to version 1.12
- Term::ReadLine upgraded to version 1.03
- Test::Builder upgraded to version 0.80
- Test::Harness upgraded version to 2.64
 - this makes it able to handle newlines.
- Test::More upgraded to version 0.80
- Test::Simple upgraded to version 0.80
- Text::Balanced upgraded to version 1.98
- Text::ParseWords upgraded to version 3.27
- Text::Soundex upgraded to version 3.03
- Text::Tabs upgraded to version 2007.1117
- Text::Wrap upgraded to version 2006.1117
- Thread upgraded to version 2.01
- Thread::Semaphore upgraded to version 2.09
- Thread::Queue upgraded to version 2.11
 - added capability to add complex structures (e.g., hash of hashes) to queues.
 - added capability to dequeue multiple items at once.
 - added new methods to inspect and manipulate queues: peek, insert and extract
- Tie::Handle upgraded to version 4.2
- Tie::Hash upgraded to version 1.03
- Tie::Memoize upgraded to version 1.1
 - Tie::Memoize::EXISTS now correctly caches its results.
- Tie::RefHash upgraded to version 1.38
- Tie::Scalar upgraded to version 1.01
- Tie::StdHandle upgraded to version 4.2
- Time::gmtime upgraded to version 1.03
- Time::Local upgraded to version 1.1901
- Time::HiRes upgraded to version 1.9715 with various build improvements (including VMS) and minor platform-specific bug fixes (including for HP-UX 11 ia64).
- threads upgraded to 1.71
 - new thread state information methods: is_running, is_detached and is_joinable. list method enhanced to return running or joinable threads.
 - new thread signal method: kill
 - added capability to specify thread stack size.

- added capability to control thread exiting behavior. Added a new `exit` method.
- `threads::shared` upgraded to version 1.27
 - smaller and faster implementation that eliminates one internal structure and the consequent level of indirection.
 - user locks are now stored in a safer manner.
 - new function `shared_clone` creates a copy of an object leaving shared elements as-is and deep-cloning non-shared elements.
 - added new `is_shared` method.
- `Unicode::Normalize` upgraded to version 1.02
- `Unicode::UCD` upgraded to version 0.25
- `warnings` upgraded to version 1.05_01
- `Win32` upgraded to version 0.38
 - added new function `GetCurrentProcessId` which returns the regular Windows process identifier of the current process, even when called from within a fork.
- `XSLoader` upgraded to version 0.10
- `XS::APITest` and `XS::Typemap` are for internal use only and hence no longer installed. Many more tests have been added to `XS::APITest`.

Utility Changes

debugger upgraded to version 1.31

- Andreas König contributed two functions to save and load the debugger history.
- `NEXT::AUTOLOAD` no longer emits warnings under the debugger.
- The debugger should now correctly find tty the device on OS X 10.5 and VMS when the program forks.
- `LVALUE` subs now work inside the debugger.

perlthanks

Perl 5.8.9 adds a new utility *perlthanks*, which is a variant of *perlbug*, but for sending non-bug-reports to the authors and maintainers of Perl. Getting nothing but bug reports can become a bit demoralising - we'll see if this changes things.

perlbug

perlbug now checks if you're reporting about a non-core module and suggests you report it to the CPAN author instead.

h2xs

- won't define an empty string as a constant [RT #25366]
- has examples for `h2xs -X`

h2ph

- now attempts to deal sensibly with the difference in path implications between `" "` and `<>` quoting in `#include` statements.
- now generates correct code for `#if defined A || defined B` [RT #39130]

New Documentation

As usual, the documentation received its share of corrections, clarifications and other nitfixes. More tags were added for indexing.

perlunitut is a tutorial written by Juero Waalboer on Unicode-related terminology and how to correctly handle Unicode in Perl scripts.

perlunicode is updated in section user defined properties.

perluniintro has been updated in the example of detecting data that is not valid in particular encoding.

perlcommunity provides an overview of the Perl Community along with further resources.

CORE documents the pseudo-namespace for Perl's core routines.

Changes to Existing Documentation

perlglossary adds *deprecated modules and features* and *to be dropped modules*.

perlhack has been updated and added resources on smoke testing.

The Perl FAQs (*perlfaq1..perlfaq9*) have been updated.

perlcheat is updated with better details on `\w`, `\d`, and `\s`.

perldebug is updated with information on how to call the debugger.

perldiag documentation updated with *subroutine with an ampersand* on the argument to `exists` and `delete` and also several terminology updates on warnings.

perlfork documents the limitation of `exec` inside pseudo-processes.

perlfunc:

- Documentation is fixed in section `caller` and `pop`.
- Function `alarm` now mentions `Time::HiRes::ualarm` in preference to `select`.
- Regarding precedence in `-x`, filetest operators are the same as unary operators, but not regarding parsing and parentheses (spotted by Eirik Berg Hanssen).
- *reverse* function documentation received scalar context examples.

perllocale documentation is adjusted for number localization and `POSIX::setlocale` to fix Debian bug #379463.

perlmodlib is updated with `CPAN::API::HOWTO` and `Sys::Syslog::win32::Win32`

perlre documentation updated to reflect the differences between `[[:xxxxx:]]` and `\p{IsXxxxx}` matches. Also added section on `/g` and `/c` modifiers.

perlreguts describe the internals of the regular expressions engine. It has been contributed by Yves Orton.

perlrebackslash describes all perl regular expression backslash and escape sequences.

perlrecharclass describes the syntax and use of character classes in Perl Regular Expressions.

perlrun is updated to clarify on the hash seed `PERL_HASH_SEED`. Also more information in options `-x` and `-u`.

perlsub example is updated to use a lexical variable for `opendir` syntax.

perlvar fixes confusion about real GID `$ (` and effective GID `$)`.

Perl thread tutorial example is fixed in section "*Queues: Passing Data Around*" in *perlthrtut* and *perlothrtut*.

perlhack documentation extensively improved by Jarkko Hietaniemi and others.

perltoot provides information on modifying `@UNIVERSAL::ISA`.

perlport documentation extended to include different `kill(-9, ...)` semantics on Windows. It also clearly states `dump` is not supported on Win32 and cygwin.

INSTALL has been updated and modernised.

Performance Enhancements

- The default since perl 5.000 has been for perl to create an empty scalar with every new typeglob. The increased use of lexical variables means that most are now unused. Thanks to Nicholas Clark's efforts, Perl can now be compiled with `-DPERL_DONT_CREATE_GVSV` to avoid creating these empty scalars. This will significantly decrease the number of scalars allocated for all configurations, and the number of scalars that need to be copied for ithread creation. Whilst this option is binary compatible with existing perl installations, it does change a long-standing assumption about the internals, hence it is not enabled by default, as some third party code may rely on the old behaviour.

We would recommend testing with this configuration on new deployments of perl, particularly for multi-threaded servers, to see whether all third party code is compatible with it, as this configuration may give useful performance improvements. For existing installations we would not recommend changing to this configuration unless thorough testing is performed before deployment.

- `diagnostics` no longer uses `$&`, which results in large speedups for regexp matching in all code using it.
- Regular expressions 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)
- Creating anonymous array and hash references (ie. `[]` and `{}`) now incurs no more overhead than creating an anonymous list or hash. Nicholas Clark provided changes with a saving of two ops and one stack push, which was measured as a slightly better than 5% improvement for these operations.
- Many calls to `strlen()` have been eliminated, either because the length was already known, or by adopting or enhancing APIs that pass lengths. This has been aided by the adoption of a `my_sprintf()` wrapper, which returns the correct C89 value - the length of the formatted string. Previously we could not rely on the return value of `sprintf()`, because on some ancient but extant platforms it still returns `char *`.
- `index` is now faster if the search string is stored in UTF-8 but only contains characters in the Latin-1 range.
- The Unicode swatch cache inside the regexp engine is now used. (the lookup had a key mismatch, present since the initial implementation). [RT #42839]

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.

At start time, if any paths in `@INC` or `Config` that *Configure* marked as relocatable (by starting them with `".../"`), then they are prefixed the directory of `$^X`. This allows 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.

Configuration improvements

Configure is now better at removing temporary files. Tom Callaway (from RedHat) also contributed patches that complete the set of flags passed to the compiler and the linker, in particular that `-fPIC` is now enabled on Linux. It will also croak when your `/dev/null` isn't a device.

A new configuration variable `d_pseudofork` has been to *Configure*, and is available as `$Config{d_pseudofork}` in the *Config* module. This distinguishes real `fork` support from the pseudofork emulation used on Windows platforms.

Config.pod and *config.sh* are now placed correctly for cross-compilation.

`$Config{useshrplib}` is now 'true' rather than 'yes' when using a shared perl library.

Compilation improvements

Parallel makes should work properly now, although there may still be problems if `make test` is instructed to run in parallel.

Many compilation warnings have been cleaned up. A very stubborn compiler warning in `S_emulate_eaccess()` was killed after six attempts. `g++` support has been tuned, especially for FreeBSD.

mkppport has been integrated, and all *ppport.h* files in the core will now be autogenerated at build time (and removed during cleanup).

Installation improvements.

installman now works with `-Duserelocatableinc` and `DESTDIR`.

installperl no longer installs:

- static library files of statically linked extensions when a shared perl library is being used. (They are not needed. See *Windows* below).
- *SIGNATURE* and *PAUSE*.pub* (CPAN files)
- *NOTES* and *PATCHING* (ExtUtils files)
- *perlld* and *ld2* (Cygwin files)

Platform Specific Changes

There are improved hints for AIX, Cygwin, DEC/OSF, FreeBSD, HP/UX, Irix 6 Linux, MachTen, NetBSD, OS/390, QNX, SCO, Solaris, SunOS, System V Release 5.x (UnixWare 7, OpenUNIX 8), Ultrix, UMIPS, uts and VOS.

FreeBSD

- Drop `-std=c89` and `-ansi` if using `long long` as the main integral type, else in FreeBSD 6.2 (and perhaps other releases), system headers do not declare some functions required by perl.

Solaris

- Starting with Solaris 10, we do not want versioned shared libraries, because those often indicate a private use only library. These problems could often be triggered when *SUNWbdb* (Berkeley DB) was installed. Hence if Solaris 10 is detected set `ignore_versioned_solibs=y`.

VMS

- Allow IEEE math to be deselected on OpenVMS I64 (but it remains the default).
- Record IEEE usage in `config.h`
- Help older VMS compilers by using `ccflags` when building `munchconfig.exe`.
- Don't try to build old `Thread` extension on VMS when `-Duseithreads` has been chosen.
- Passing a raw string of "NaN" to `nawk` causes a core dump - so the string has been changed to `"*NaN"`
- `t/op/stat.t` tests will now test hard links on VMS if they are supported.

Windows

- When using a shared perl library `installperl` no longer installs static library files, import library files and export library files (of statically linked extensions) and empty bootstrap files (of dynamically linked extensions). This fixes a problem building PAR-Packer on Win32 with a debug build of perl.
- Various improvements to the win32 build process, including support for Visual C++ 2005 Express Edition (aka Visual C++ 8.x).
- `perl.exe` will now have an icon if built with MinGW or Borland.
- Improvements to the `perl-static.exe` build process.
- Add Win32 makefile option to link all extensions statically.
- The `WinCE` directory has been merged into the `Win32` directory.
- `setlocale` tests have been re-enabled for Windows XP onwards.

Selected Bug Fixes

Unicode

Many many bugs related to the internal Unicode implementation (UTF-8) have been fixed. In particular, long standing bugs related to returning Unicode via `tie`, overloading or `$@` are now gone, some of which were never reported.

`unpack` will internally convert the string back from UTF-8 on numeric types. This is a compromise between the full consistency now in 5.10, and the current behaviour, which is often used as a "feature" on string types.

Using `:crlf` and UTF-16 IO layers together will now work.

Fixed problems with `split`, Unicode `/\s+/` and `/\0/`.

Fixed bug RT #40641 - encoding of Unicode characters in regular expressions.

Fixed a bug where using certain patterns in a regexp led to a panic. [RT #45337]

Perl no longer segfaults (due to infinite internal recursion) if the locale's character is not UTF-8 [RT #41442]:

```
use open ':locale';
print STDERR "\x{201e}"; # &bdquo;
```

PerlIO

Inconsistencies have been fixed in the reference counting PerlIO uses to keep track of Unix file descriptors, and the API used by XS code to manage getting and releasing `FILE *s`

Magic

Several bugs have been fixed in Magic, the internal system used to implement features such as `tie`, tainting and threads sharing.

`undef @array` on a tied array now correctly calls the `CLEAR` method.

Some of the bitwise ops were not checking whether their arguments were magical before using them. [RT #24816]

Magic is no longer invoked twice by the expression `\&$x`

A bug with assigning large numbers and tainting has been resolved. [RT #40708]

A new entry has been added to the MAGIC vtable - `svt_local`. This is used when copying magic to the new value during `local`, allowing certain problems with localising shared variables to be resolved.

For the implementation details, see *"Magic Virtual Tables" in perl guts*.

Reblessing overloaded objects now works

Internally, perl object-ness is on the referent, not the reference, even though methods can only be called via a reference. However, the original implementation of overloading stored flags related to overloading on the reference, relying on the flags being copied when the reference was copied, or set at the creation of a new reference. This manifests in a bug - if you rebless an object from a class that has overloading, into one that does not, then any other existing references think that they (still) point to an overloaded object, choose these C code paths, and then throw errors. Analogously, blessing into an overloaded class when other references exist will result in them not using overloading.

The implementation has been fixed for 5.10, but this fix changes the semantics of flag bits, so is not binary compatible, so can't be applied to 5.8.9. However, 5.8.9 has a work-around that implements the same bug fix. If the referent has multiple references, then all the other references are located and corrected. A full search is avoided whenever possible by scanning lexicals outwards from the current subroutine, and the argument stack.

A certain well known Linux vendor applied incomplete versions of this bug fix to their `/usr/bin/perl` and then prematurely closed bug reports about performance issues without consulting back upstream. This not being enough, they then proceeded to ignore the necessary fixes to these unreleased changes for 11 months, until massive pressure was applied by their long-suffering paying customers, catalysed by the failings being featured on a prominent blog and Slashdot.

strict now propagates correctly into string evals

Under 5.8.8 and earlier:

```
$ perl5.8.8 -e 'use strict; eval "use foo bar" or die $@'
Can't locate foo.pm in @INC (@INC contains: ... .) at (eval 1) line 2.
BEGIN failed--compilation aborted at (eval 1) line 2.
```

Under 5.8.9 and later:

```
$ perl5.8.9 -e 'use strict; eval "use foo bar" or die $@'
Bareword "bar" not allowed while "strict subs" in use at (eval 1) line
1.
```

This may cause problems with programs that parse the error message and rely on the buggy behaviour.

Other fixes

- The tokenizer no longer treats `=cute` (and other words beginning with `=cut`) as a synonym for `=cut`.
- Calling `CORE::require`
`CORE::require` and `CORE::do` were always parsed as `require` and `do` when they were overridden. This is now fixed.
- Stopped memory leak on long `/etc/groups` entries.
- `while (my $x ...) { ...; redo } shouldn't undef $x`.
In the presence of `my` in the conditional of a `while()`, `until()`, or `for(;;)` loop, we now add an extra scope to the body so that `redo` doesn't undef the lexical.
- The `encoding` pragma now correctly ignores anything following an `@` character in the `LC_ALL` and `LANG` environment variables. [RT # 49646]
- A segfault observed with some `gcc 3.3` optimisations is resolved.
- A possible segfault when `unpack` used in scalar context with `()` groups is resolved. [RT #50256]
- Resolved issue where `$!` could be changed by a signal handler interrupting a `system` call.
- Fixed bug RT #37886, symbolic deferencing was allowed in the argument of `defined` even under the influence of `use strict 'refs'`.
- Fixed bug RT #43207, where `lc/uc` inside `sort` affected the return value.
- Fixed bug RT #45607, where `*{"BONK"} = \&{"BONK"}` didn't work correctly.
- Fixed bug RT #35878, croaking from a XSUB called via `goto &xsub` corrupts perl internals.
- Fixed bug RT #32539, `DynaLoader.o` is moved into `libperl.so` to avoid the need to statically link `DynaLoader` into the stub perl executable. With this `libperl.so` provides everything needed to get a functional embedded perl interpreter to run.
- Fix bug RT #36267 so that assigning to a tied hash doesn't change the underlying hash.
- Fix bug RT #6006, `regexp` replaces using large replacement variables fail some of the time, *i.e.* when substitution contains something like `${10}` (note the bracket) instead of just `$10`.
- Fix bug RT #45053, `Perl_newCONSTSUB()` is now thread safe.

Platform Specific Fixes

Darwin / MacOS X

- Various improvements to 64 bit builds.
- Mutex protection added in `PerlIOStdio_close()` to avoid race conditions. Hopefully this fixes failures in the threads tests `free.t` and `blocks.t`.
- Added forked terminal support to the debugger, with the ability to update the window title.

OS/2

- A build problem with specifying `USE_MULTI` and `USE_ITHREADS` but without `USE_IMP_SYS` has been fixed.
- `OS2::REXX` upgraded to version 1.04

Tru64

- Aligned floating point build policies for *cc* and *gcc*.

RedHat Linux

- Revisited a patch from 5.6.1 for RH7.2 for Intel's *icc* [RT #7916], added an additional check for `$Config{gccversion}`.

Solaris/i386

- Use `-DPTR_IS_LONG` when using 64 bit integers

VMS

- Fixed `PerlIO::Scalar` in-memory file record-style reads.
- pipe shutdown at process exit should now be more robust.
- Bugs in VMS exit handling tickled by `Test::Harness 2.64` have been fixed.
- Fix `fcntl()` locking capability test in *configure.com*.
- Replaced `shrplib='define'` with `useshrplib='true'` on VMS.

Windows

- `File::Find` used to fail when the target directory is a bare drive letter and `no_chdir` is 1 (the default is 0). [RT #41555]
- A build problem with specifying `USE_MULTI` and `USE_ITHREADS` but without `USE_IMP_SYS` has been fixed.
- The process id is no longer truncated to 16 bits on some Windows platforms (http://bugs.activestate.com/show_bug.cgi?id=72443)
- Fixed bug RT #54828 in *perlio.c* where calling `binmode` on Win32 and Cgywin may cause a segmentation fault.

Smaller fixes

- It is now possible to overload `eq` when using `nomethod`.
- Various problems using `overload` with 64 bit integers corrected.
- The reference count of `PerlIO` file descriptors is now correctly handled.
- On VMS, escaped dots will be preserved when converted to UNIX syntax.
- `keys %+` no longer throws an 'ambiguous' warning.
- Using `#!perl -d` could trigger an assertion, which has been fixed.
- Don't stringify tied code references in `@INC` when calling `require`.
- Code references in `@INC` report the correct file name when `__FILE__` is used.
- Width and precision in `sprintf` didn't handle characters above 255 correctly. [RT #40473]
- List slices with indices out of range now work more consistently. [RT #39882]
- A change introduced with perl 5.8.1 broke the parsing of arguments of the form `-foo=bar` with the `-s` on the `<#!>` line. This has been fixed. See http://bugs.activestate.com/show_bug.cgi?id=43483
- `tr///` is now threadsafe. Previously it was storing a swash inside its OP, rather than in a pad.
- *pod2html* labels anchors more consistently and handles nested definition lists better.

- threads cleanup veto has been extended to include `perl_free()` and `perl_destruct()`
- On some systems, changes to `$ENV{TZ}` would not always be respected by the underlying calls to `localtime_r()`. Perl now forces the inspection of the environment on these systems.
- The special variable `$_R` is now more consistently set when executing regexps using the `(?{...})` construct. In particular, it will still be set even if backreferences or optional sub-patterns `(?:...)?` are used.

New or Changed Diagnostics

panic: sv_chop %s

This new fatal error occurs when the C routine `Perl_sv_chop()` was passed a position that is not within the scalar's string buffer. This is caused by buggy XS code, and at this point recovery is not possible.

Maximal count of pending signals (%s) exceeded

This new fatal error occurs when the perl process has to abort due to too many pending signals, which is bound to prevent perl from being able to handle further incoming signals safely.

panic: attempt to call %s in %s

This new fatal error occurs when the ACL version file test operator is used where it is not available on the current platform. Earlier checks mean that it should never be possible to get this.

FETCHSIZE returned a negative value

New error indicating that a tied array has claimed to have a negative number of elements.

Can't upgrade %s (%d) to %d

Previously the internal error from the SV upgrade code was the less informative *Can't upgrade that kind of scalar*. It now reports the current internal type, and the new type requested.

%s argument is not a HASH or ARRAY element or a subroutine

This error, thrown if an invalid argument is provided to `exists` now correctly includes "or a subroutine". [RT #38955]

Cannot make the non-overridable builtin %s fatal

This error in `Fatal` previously did not show the name of the builtin in question (now represented by %s above).

Unrecognized character '%s' in column %d

This error previously did not state the column.

Offset outside string

This can now also be generated by a `seek` on a file handle using `PerlIO::scalar`.

Invalid escape in the specified encoding in regexp; marked by <-- HERE in m/%s/

New error, introduced as part of the fix to RT #40641 to handle encoding of Unicode characters in regular expression comments.

Your machine doesn't support dump/undump.

A more informative fatal error issued when calling `dump` on Win32 and Cygwin. (Given that the purpose of `dump` is to abort with a core dump, and core dumps can't be produced on these platforms, this is more useful than silently exiting.)

Changed Internals

The perl sources can now be compiled with a C++ compiler instead of a C compiler. A necessary implementation details is that under C++, the macro `XS` used to define XSUBs now includes an `extern "C"` definition. A side effect of this is that **C++** code that used the construction

```
typedef XS(SwigPerlWrapper);
```

now needs to be written

```
typedef XSPROTO(SwigPerlWrapper);
```

using the new `XSPROTO` macro, in order to compile. C extensions are unaffected, although C extensions are encouraged to use `XSPROTO` too. This change was present in the 5.10.0 release of perl, so any actively maintained code that happened to use this construction should already have been adapted. Code that needs changing will fail with a compilation error.

set magic on localizing/assigning to a magic variable will now only trigger for *container magics*, i.e. it will for `%ENV` or `%SIG` but not for `$#array`.

The new API macro `newSVpvs()` can be used in place of constructions such as `newSVpv("ISA", 3)`. It takes a single string constant, and at C compile time determines its length.

The new API function `Perl_newSV_type()` can be used as a more efficient replacement of the common idiom

```
sv = newSV(0);
sv_upgrade(sv, type);
```

Similarly `Perl_newSVpv_flags()` can be used to combine `Perl_newSVpv()` with `Perl_sv_2mortal()` or the equivalent `Perl_sv_newmortal()` with `Perl_sv_setpv()`

Two new macros `mPUSHs()` and `mXPUSHs()` are added, to make it easier to push mortal SVs onto the stack. They were then used to fix several bugs where values on the stack had not been mortalised.

A `Perl_signbit()` function was added to test the sign of an NV. It maps to the system one when available.

`Perl_av_reify()`, `Perl_lex_end()`, `Perl_mod()`, `Perl_op_clear()`, `Perl_pop_return()`, `Perl_qerror()`, `Perl_setdefout()`, `Perl_vivify_defelem()` and `Perl_yylex()` are now visible to extensions. This was required to allow `Data::Alias` to work on Windows.

`Perl_find_runcv()` is now visible to perl core extensions. This was required to allow `Sub::Current` to work on Windows.

`ptr_table*` functions are now available in unthreaded perl. `Storable` takes advantage of this.

There have been many small cleanups made to the internals. In particular, `Perl_sv_upgrade()` has been simplified considerably, with a straight-through code path that uses `memset()` and `memcpy()` to initialise the new body, rather than assignment via multiple temporary variables. It has also benefited from simplification and de-duplication of the arena management code.

A lot of small improvements in the code base were made due to reports from the Coverity static code analyzer.

Corrected use and documentation of `Perl_gv_stashpv()`, `Perl_gv_stashpv()`, `Perl_gv_stashsv()` functions (last parameter is a bitmask, not boolean).

`PERL_SYS_INIT`, `PERL_SYS_INIT3` and `PERL_SYS_TERM` macros have been changed into functions.

PERLSYS_TERM no longer requires a context. `PerlIO_teardown()` is now called without a context, and debugging output in this function has been disabled because that required that an interpreter was present, an invalid assumption at termination time.

All compile time options which affect binary compatibility have been grouped together into a global variable (`PL_bincompat_options`).

The values of `PERL_REVISION`, `PERL_VERSION` and `PERL_SUBVERSION` are now baked into global variables (and hence into any shared perl library). Additionally under `MULTIPLICITY`, the perl executable now records the size of the interpreter structure (total, and for this version). Coupled with `PL_bincompat_options` this will allow 5.8.10 (and later), when compiled with a shared perl library, to perform sanity checks in `main()` to verify that the shared library is indeed binary compatible.

Symbolic references can now have embedded NULs. The new public function `Perl_get_cvn_flags()` can be used in extensions if you have to handle them.

Macro cleanups

The core code, and XS code in *ext* that is not dual-lived on CPAN, no longer uses the macros `PL_na`, `NEWSV()`, `Null()`, `Nullav`, `Nullcv`, `Nullhv`, `Nullhv` etc. Their use is discouraged in new code, particularly `PL_na`, which is a small performance hit.

New Tests

Many modules updated from CPAN incorporate new tests. Some core specific tests have been added:

`ext/DynaLoader/t/DynaLoader.t`

Tests for the `DynaLoader` module.

`t/comp/fold.t`

Tests for compile-time constant folding.

`t/io/pvbm.t`

Tests incorporated from 5.10.0 which check that there is no unexpected interaction between the internal types `PVBM` and `PVGV`.

`t/lib/proxy_constant_subs.t`

Tests for the new form of constant subroutines.

`t/op/attrhand.t`

Tests for `Attribute::Handlers`.

`t/op/dbm.t`

Tests for `dbmopen`.

`t/op/inccode-tie.t`

Calls all tests in `t/op/inccode.t` after first tying `@INC`.

`t/op/incfilter.t`

Tests for for source filters returned from code references in `@INC`.

`t/op/kill0.t`

Tests for RT #30970.

`t/op/qstack.t`

Tests for RT #41484.

`t/op/qr.t`

Tests for the `qr//` construct.

`t/op/regexp_qr_embed.t`

Tests for the `qr//` construct within another regexp.

`t/op/regexp_qr.t`

Tests for the `qr//` construct.

`t/op/rxcode.t`

Tests for RT #32840.

`t/op/study_tied.t`

Tests for `study` on tied scalars.

`t/op/substT.t`

Tests for `subst` run under `-T` mode.

`t/op/symbolcache.t`

Tests for `undef` and `delete` on stash entries that are bound to subroutines or methods.

`t/op/upgrade.t`

Tests for `Perl_sv_upgrade()`.

`t/mro/package_aliases.t`

MRO tests for `isa` and package aliases.

`t/pod/twice.t`

Tests for calling `Pod::Parser` twice.

`t/run/cloexec.t`

Tests for inheriting file descriptors across `exec` (close-on-exec).

`t/uni/cache.t`

Tests for the UTF-8 caching code.

`t/uni/chr.t`

Test that strange encodings do not upset `Perl_pp_chr()`.

`t/uni/greek.t`

Tests for RT #40641.

`t/uni/latin2.t`

Tests for RT #40641.

`t/uni/overload.t`

Tests for returning Unicode from overloaded values.

`t/uni/tie.t`

Tests for returning Unicode from tied variables.

Known Problems

There are no known new bugs.

However, programs that rely on bugs that have been fixed will have problems. Also, many bug fixes present in 5.10.0 can't be back-ported to the 5.8.x branch, because they require changes that are binary incompatible, or because the code changes are too large and hence too risky to incorporate.

We have only limited volunteer labour, and the maintenance burden is getting increasingly complex. Hence this will be the last significant release of the 5.8.x series. Any future releases of 5.8.x will likely only be to deal with security issues, and platform build failures. Hence you should look to migrating to 5.10.x, if you have not started already. Alternatively, if business requirements constrain you to continue to use 5.8.x, you may wish to consider commercial support from firms such as ActiveState.

Platform Specific Notes

Win32

`readdir()`, `cwd()`, `$^X` and `@INC` now use the alternate (short) filename if the long name is outside the current codepage (Jan Dubois).

Updated Modules

- `Win32` upgraded to version 0.38. Now has a documented 'WinVista' response from `GetOSName` and support for Vista's privilege elevation in `IsAdminUser`. Support for Unicode characters in path names. Improved `cygwin` and `Win64` compatibility.
- `Win32API` updated to 0.1001_01
- `killpg()` support added to `MSWin32` (Jan Dubois).
- `File::Spec::Win32` upgraded to version 3.2701

OS/2

Updated Modules

- `OS2::Process` upgraded to 1.03
Ilya Zakharevich has added and documented several `Window*` and `Clipbrd*` functions.
- `OS2::REXX::DLL`, `OS2::REXX` updated to version 1.03

VMS

Updated Modules

- `DCLsym` upgraded to version 1.03
- `Stdio` upgraded to version 2.4
- `VMS::XSSymSet` upgraded to 1.1.

Obituary

Nick Ing-Simmons, long time Perl hacker, author of the `Tk` and `Encode` modules, *perlio.c* in the core, and 5.003_02 pumpking, died of a heart attack on 25th September 2006. He will be missed.

Acknowledgements

Some of the work in this release was funded by a TPF grant.

Steve Hay worked behind the scenes working out the causes of the differences between core modules, their CPAN releases, and previous core releases, and the best way to rectify them. He doesn't want to do it again. I know this feeling, and I'm very glad he did it this time, instead of me.

Paul Fenwick assembled a team of 18 volunteers, who broke the back of writing this document. In particular, Bradley Dean, Eddy Tan, and Vincent Pit provided half the team's contribution.

Schwern verified the list of updated module versions, correcting quite a few errors that I (and everyone else) had missed, both wrongly stated module versions, and changed modules that had not been listed.

The crack Berlin-based QA team of Andreas König and Slaven Rezić tirelessly re-built snapshots, tested most everything CPAN against them, and then identified the changes responsible for any module regressions, ensuring that several show-stopper bugs were stomped before the first release

candidate was cut.

The other core committers contributed most of the changes, and applied most of the patches sent in by the hundreds of contributors listed in *AUTHORS*.

And obviously, Larry Wall, without whom we wouldn't have Perl.

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. You can browse and search the Perl 5 bugs at <http://bugs.perl.org/>

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 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.