NAME

perl570delta - what's new for perl v5.7.0

DESCRIPTION

This document describes differences between the 5.6.0 release and the 5.7.0 release.

Security Vulnerability Closed

A potential security vulnerability in the optional suidperl component of Perl has been identified. suidperl is neither built nor installed by default. As of September the 2nd, 2000, the only known vulnerable platform is Linux, most likely all Linux distributions. CERT and various vendors have been alerted about the vulnerability.

The problem was caused by Perl trying to report a suspected security exploit attempt using an external program, /bin/mail. On Linux platforms the /bin/mail program had an undocumented feature which when combined with suidperl gave access to a root shell, resulting in a serious compromise instead of reporting the exploit attempt. If you don't have /bin/mail, or if you have 'safe setuid scripts', or if suidperl is not installed, you are safe.

The exploit attempt reporting feature has been completely removed from the Perl 5.7.0 release, so that particular vulnerability isn't there anymore. However, further security vulnerabilities are, unfortunately, always possible. The suidperl code is being reviewed and if deemed too risky to continue to be supported, it may be completely removed from future releases. In any case, suidperl should only be used by security experts who know exactly what they are doing and why they are using suidperl instead of some other solution such as sudo (see http://www.courtesan.com/sudo/).

Incompatible Changes

- Arrays now always interpolate into double-quoted strings: constructs like "foo@bar" now always assume @bar is an array, whether or not the compiler has seen use of @bar.

- The semantics of bless(REF, REF) were unclear and until someone proves it to make some sense, it is forbidden.

- A reference to a reference now stringify as "REF(0x81485ec)" instead of "SCALAR(0x81485ec)" in order to be more consistent with the return value of ref().

- The very dusty examples in the eg/ directory have been removed. Suggestions for new shiny examples welcome but the main issue is that the examples need to be documented, tested and (most importantly) maintained.

- The obsolete chat2 library that should never have been allowed to escape the laboratory has been decommissioned.

- The unimplemented POSIX regex features [[.cc.]] and [[=c=]] are still recognised but now cause fatal errors. The previous behaviour of ignoring them by default and warning if requested was unacceptable since it, in a way, falsely promised that the features could be used.

- The (bogus) escape sequences \8 and \9 now give an optional warning ("Unrecognized escape passed through"). There is no need to \-escape any \w character.

- lstat(FILEHANDLE) now gives a warning because the operation makes no sense. In future releases this may become a fatal error.

- The long deprecated uppercase aliases for the string comparison operators (EQ, NE, LT, LE, GE, GT) have now been removed.

- The regular expression captured submatches ($1, $2, ...) are now more consistently unset if the match fails, instead of leaving false data lying around in them.

- The tr///C and tr///U features have been removed and will not return; the interface was a

mistake. Sorry about that. For similar functionality, see pack('U0', ...) and pack('C0', ...).

Core Enhancements

- `perl -d:Module=arg,arg,arg` now works (previously one couldn't pass in multiple arguments.)

- my __PACKAGE__ $obj now works.

- `no Module;` now works even if there is no "sub unimport" in the Module.

- The numerical comparison operators return `undef` if either operand is a NaN. Previously the behaviour was unspecified.

- `pack('U0a*', ...)` can now be used to force a string to UTF-8.

- prototype(\&) is now available.

- There is now an UNTIE method.

Modules and Pragmata

New Modules

- File::Temp allows one to create temporary files and directories in an easy, portable, and secure way.

- Storable gives persistence to Perl data structures by allowing the storage and retrieval of Perl data to and from files in a fast and compact binary format.

Updated And Improved Modules and Pragmata

- The following independently supported modules have been updated to newer versions from CPAN: CGI, CPAN, DB_File, File::Spec, Getopt::Long, the podlators bundle, Pod::LaTeX, Pod::Parser, Term::ANSIColor, Test.

- Bug fixes and minor enhancements have been applied to B::Deparse, Data::Dumper, IO::Poll, IO::Socket::INET, Math::BigFloat, Math::Complex, Math::Trig, Net::protoent, the re pragma, SelfLoader, Sys::SysLog, Test::Harness, Text::Wrap, UNIVERSAL, and the warnings pragma.

- The attributes::reftype() now works on tied arguments.

- AutoLoader can now be disabled with `no AutoLoader;`,

- The English module can now be used without the infamous performance hit by saying

  ```
  use English '-no_performance_hit';
  ```

  (Assuming, of course, that one doesn't need the troublesome variables $`, $&, or $'.) Also, introduced @LAST_MATCH_START and @LAST_MATCH_END English aliases for @- and @+.

- File::Find now has pre- and post-processing callbacks. It also correctly changes directories when chasing symbolic links. Callbacks (naughtily) exiting with "next;" instead of "return;" now work.

- File::Glob::glob() renamed to File::Glob::bsd_glob() to avoid prototype mismatch with CORE::glob().

- IPC::Open3 now allows the use of numeric file descriptors.

- use lib now works identically to @INC. Removing directories with 'no lib' now works.

- `%INC` now localised in a Safe compartment so that use/require work.

- The Shell module now has an OO interface.

Utility Changes

- The Emacs perl mode (emacs/cperl-mode.el) has been updated to version 4.31.

- Perlbug is now much more robust. It also sends the bug report to perl.org, not perl.com.

- The perlcc utility has been rewritten and its user interface (that is, command line) is much more like that of the Unix C compiler, cc.

- The xsubpp utility for extension writers now understands POD documentation embedded in the *.xs files.

New Documentation

- perl56delta details the changes between the 5.005 release and the 5.6.0 release.

- perldebtut is a Perl debugging tutorial.

- perlebcdic contains considerations for running Perl on EBCDIC platforms. Note that unfortunately EBCDIC platforms that used to supported back in Perl 5.005 are still unsupported by Perl 5.7.0; the plan, however, is to bring them back to the fold.

- perlnewmod tells about writing and submitting a new module.

- perlposix-bc explains using Perl on the POSIX-BC platform (an EBCDIC mainframe platform).

- perlretut is a regular expression tutorial.

- perlrequick is a regular expressions quick-start guide. Yes, much quicker than perlretut.

- perlutil explains the command line utilities packaged with the Perl distribution.

Performance Enhancements

- map() that changes the size of the list should now work faster.

- sort() has been changed to use mergesort internally as opposed to the earlier quicksort. For very small lists this may result in slightly slower sorting times, but in general the speedup should be at least 20%. Additional bonuses are that the worst case behaviour of sort() is now better (in computer science terms it now runs in time O(N log N), as opposed to quicksort's Theta(N**2) worst-case run time behaviour), and that sort() is now stable (meaning that elements with identical keys will stay ordered as they were before the sort).

Installation and Configuration Improvements

Generic Improvements

- INSTALL now explains how you can configure Perl to use 64-bit integers even on non-64-bit platforms.

- Policy.sh policy change: if you are reusing a Policy.sh file (see INSTALL) and you use Configure -Dprefix=/foo/bar and in the old Policy $prefix eq $siteprefix and $prefix eq $vendorprefix, all of them will now be changed to the new prefix, /foo/bar. (Previously only $prefix changed.) If you do not like this new behaviour, specify prefix, siteprefix, and vendorprefix explicitly.

- A new optional location for Perl libraries, otherlibdirs, is available. It can be used for example for vendor add-ons without disturbing Perl's own library directories.

- In many platforms the vendor-supplied 'cc' is too stripped-down to build Perl (basically, 'cc' doesn't do ANSI C). If this seems to be the case and 'cc' does not seem to be the GNU C compiler 'gcc', an automatic attempt is made to find and use 'gcc' instead.

- gcc needs to closely track the operating system release to avoid build problems. If Configure finds that gcc was built for a different operating system release than is running, it now gives a

clearly visible warning that there may be trouble ahead.

- If binary compatibility with the 5.005 release is not wanted, Configure no longer suggests including the 5.005 modules in @INC.

- Configure `-S` can now run non-interactively.

- configure.gnu now works with options with whitespace in them.

- installperl now outputs everything to STDERR.

- $Config{byteorder} is now computed dynamically (this is more robust with "fat binaries" where an executable image contains binaries for more than one binary platform.)

Selected Bug Fixes

- Several debugger fixes: exit code now reflects the script exit code, condition `"0"` now treated correctly, the d command now checks line number, the `$.` no longer gets corrupted, all debugger output now goes correctly to the socket if RemotePort is set.

- `*foo{FORMAT}` now works.

- Lexical warnings now propagating correctly between scopes.

- Line renumbering with eval and `#line` now works.

- Fixed numerous memory leaks, especially in eval "".

- Modulus of unsigned numbers now works (4063328477 % 65535 used to return 27406, instead of 27047).

- Some "not a number" warnings introduced in 5.6.0 eliminated to be more compatible with 5.005. Infinity is now recognised as a number.

- our() variables will not cause "will not stay shared" warnings.

- pack "Z" now correctly terminates the string with "\0".

- Fix password routines which in some shadow password platforms (e.g. HP-UX) caused getpwent() to return every other entry.

- printf() no longer resets the numeric locale to "C".

- `q(a\\b)` now parses correctly as `'a\\b'`.

- Printing quads (64-bit integers) with printf/sprintf now works without the q L ll prefixes (assuming you are on a quad-capable platform).

- Regular expressions on references and overloaded scalars now work.

- scalar() now forces scalar context even when used in void context.

- sort() arguments are now compiled in the right wantarray context (they were accidentally using the context of the sort() itself).

- Changed the POSIX character class `[[:space:]]` to include the (very rare) vertical tab character. Added a new POSIX-ish character class `[[:blank:]]` which stands for horizontal whitespace (currently, the space and the tab).

- $AUTOLOAD, sort(), lock(), and spawning subprocesses in multiple threads simultaneously are now thread-safe.

- Allow read-only string on left hand side of non-modifying tr///.

- Several Unicode fixes (but still not perfect).

- BOMs (byte order marks) in the beginning of Perl files (scripts, modules) should now be transparently skipped. UTF-16 (UCS-2) encoded Perl files should now be read correctly.

- The character tables have been updated to Unicode 3.0.1.

- chr() for values greater than 127 now create utf8 when under use utf8.

- Comparing with utf8 data does not magically upgrade non-utf8 data into utf8.

- `IsAlnum`, `IsAlpha`, and `IsWord` now match titlecase.

- Concatenation with the `.` operator or via variable interpolation, `eq`, `substr`, `reverse`, `quotemeta`, the `x` operator, substitution with `s///`, single-quoted UTF-8, should now work--in theory.

- The `tr///` operator now works *slightly* better but is still rather broken. Note that the `tr///CU` functionality has been removed (but see pack('U0', ...)).

- vec() now refuses to deal with characters >255.

- Zero entries were missing from the Unicode classes like `IsDigit`.

- UNIVERSAL::isa no longer caches methods incorrectly. (This broke the Tk extension with 5.6.0.)

Platform Specific Changes and Fixes

- BSDI 4.*

  Perl now works on post-4.0 BSD/OSes.

- All BSDs

  Setting `$0` now works (as much as possible; see perlvar for details).

- Cygwin

  Numerous updates; currently synchronised with Cygwin 1.1.4.

- EPOC

  EPOC update after Perl 5.6.0. See README.epoc.

- FreeBSD 3.*

  Perl now works on post-3.0 FreeBSDs.

- HP-UX

  README.hpux updated; `Configure -Duse64bitall` now almost works.

- IRIX

  Numerous compilation flag and hint enhancements; accidental mixing of 32-bit and 64-bit libraries (a doomed attempt) made much harder.

- Linux

  Long doubles should now work (see INSTALL).

- Mac OS Classic

  Compilation of the standard Perl distribution in Mac OS Classic should now work if you have the Metrowerks development environment and the missing Mac-specific toolkit bits. Contact the macperl mailing list for details.

- MPE/iX

MPE/iX update after Perl 5.6.0. See README.mpeix.

- NetBSD/sparc

  Perl now works on NetBSD/sparc.

- OS/2

  Now works with usethreads (see INSTALL).

- Solaris

  64-bitness using the Sun Workshop compiler now works.

- Tru64 (aka Digital UNIX, aka DEC OSF/1)

  The operating system version letter now recorded in $Config{osvers}. Allow compiling with gcc (previously explicitly forbidden). Compiling with gcc still not recommended because buggy code results, even with gcc 2.95.2.

- Unicos

  Fixed various alignment problems that lead into core dumps either during build or later; no longer dies on math errors at runtime; now using full quad integers (64 bits), previously was using only 46 bit integers for speed.

- VMS

  chdir() now works better despite a CRT bug; now works with MULTIPLICITY (see INSTALL); now works with Perl's malloc.

- Windows

    - accept() no longer leaks memory.

    - Better chdir() return value for a non-existent directory.

    - New %ENV entries now propagate to subprocesses.

    - $ENV{LIB} now used to search for libs under Visual C.

    - A failed (pseudo)fork now returns undef and sets errno to EAGAIN.

    - Allow REG_EXPAND_SZ keys in the registry.

    - Can now send() from all threads, not just the first one.

    - Fake signal handling reenabled, bugs and all.

    - Less stack reserved per thread so that more threads can run concurrently. (Still 16M per thread.)

    - `File::Spec->tmpdir()` now prefers C:/temp over /tmp (works better when perl is running as service).

    - Better UNC path handling under ithreads.

    - wait() and waitpid() now work much better.

    - winsock handle leak fixed.

New or Changed Diagnostics

All regular expression compilation error messages are now hopefully easier to understand both because the error message now comes before the failed regex and because the point of failure is now clearly marked.

The various "opened only for", "on closed", "never opened" warnings drop the `main::` prefix for

filehandles in the `main` package, for example STDIN instead of <main::STDIN>.

The "Unrecognized escape" warning has been extended to include \8, \9, and _. There is no need to escape any of the \w characters.

Changed Internals

- perlapi.pod (a companion to perlguts) now attempts to document the internal API.

- You can now build a really minimal perl called microperl. Building microperl does not require even running Configure; `make -f Makefile.micro` should be enough. Beware: microperl makes many assumptions, some of which may be too bold; the resulting executable may crash or otherwise misbehave in wondrous ways. For careful hackers only.

- Added rsignal(), whichsig(), do_join() to the publicised API.

- Made possible to propagate customised exceptions via croak()ing.

- Added is_utf8_char(), is_utf8_string(), bytes_to_utf8(), and utf8_to_bytes().

- Now xsubs can have attributes just like subs.

Known Problems

Unicode Support Still Far From Perfect

We're working on it. Stay tuned.

EBCDIC Still A Lost Platform

The plan is to bring them back.

Building Extensions Can Fail Because Of Largefiles

Certain extensions like mod_perl and BSD::Resource are known to have issues with `largefiles`, a change brought by Perl 5.6.0 in which file offsets default to 64 bits wide, where supported. Modules may fail to compile at all or compile and work incorrectly. Currently there is no good solution for the problem, but Configure now provides appropriate non-largefile ccflags, ldflags, libswanted, and libs in the %Config hash (e.g., $Config{ccflags_nolargefiles}) so the extensions that are having problems can try configuring themselves without the largefileness. This is admittedly not a clean solution, and the solution may not even work at all. One potential failure is whether one can (or, if one can, whether it's a good idea) link together at all binaries with different ideas about file offsets, all this is platform-dependent.

ftmp-security tests warn 'system possibly insecure'

Don't panic. Read INSTALL 'make test' section instead.

Test lib/posix Subtest 9 Fails In LP64-Configured HP-UX

If perl is configured with -Duse64bitall, the successful result of the subtest 10 of lib/posix may arrive before the successful result of the subtest 9, which confuses the test harness so much that it thinks the subtest 9 failed.

Long Doubles Still Don't Work In Solaris

The experimental long double support is still very much so in Solaris. (Other platforms like Linux and Tru64 are beginning to solidify in this area.)

Linux With Sfio Fails op/misc Test 48

No known fix.

Storable tests fail in some platforms

If any Storable tests fail the use of Storable is not advisable.

- Many Storable tests fail on AIX configured with 64 bit integers.

So far unidentified problems break Storable in AIX if Perl is configured to use 64 bit integers. AIX in 32-bit mode works and other 64-bit platforms work with Storable.

- DOS DJGPP may hang when testing Storable.

- st-06compat fails in UNICOS and UNICOS/mk.

    This means that you cannot read old (pre-Storable-0.7) Storable images made in other platforms.

- st-store.t and st-retrieve may fail with Compaq C 6.2 on OpenVMS Alpha 7.2.

Threads Are Still Experimental

Multithreading is still an experimental feature. Some platforms emit the following message for lib/thr5005

```
#
# This is a KNOWN FAILURE, and one of the reasons why threading
# is still an experimental feature.  It is here to stop people
# from deploying threads in production. ;-)
#
```

and another known thread-related warning is

```
pragma/overload......Unbalanced saves: 3 more saves than restores
panic: magic_mutexfree during global destruction.
ok
lib/selfloader.......Unbalanced saves: 3 more saves than restores
panic: magic_mutexfree during global destruction.
ok
lib/st-dclone........Unbalanced saves: 3 more saves than restores
panic: magic_mutexfree during global destruction.
ok
```

The Compiler Suite Is Still Experimental

The compiler suite is slowly getting better but is nowhere near working order yet. The backend part that has seen perhaps the most progress is the bytecode compiler.

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.com/perl/ , 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.

HISTORY

Written by Jarkko Hietaniemi *<jhi@iki.fi>*, with many contributions from The Perl Porters and Perl Users submitting feedback and patches.

Send omissions or corrections to *<perlbug@perl.org>*.