Rocky Enterprise Linux 9.2 Manual Pages on command 'Sort::Versions.3pm'

C:\>man Sort::Versions.3pm

Sort::Versions(3pm) User Contributed Perl Documentation Sort::Versions(3pm)

NAME

Sort::Versions - a perl 5 module for sorting of revision-like numbers

SYNOPSIS

```
use Sort::Versions;
@l = sort { versioncmp($a, $b) } qw( 1.2 1.2.0 1.2a.0 1.2.a 1.a 02.a );
...
use Sort::Versions;
print 'lower' if versioncmp('1.2', '1.2a') == -1;
...
use Sort::Versions;
%h = (1 => 'd', 2 => 'c', 3 => 'b', 4 => 'a');
@h = sort { versioncmp($h{$a}, $h{$b}) } keys %h;
```

DESCRIPTION

Sort::Versions allows easy sorting of mixed non-numeric and numeric strings, like the 'version numbers' that many shared library systems and revision control packages use. This is quite useful if you are trying to deal with shared libraries. It can also be applied to applications that intersperse variable-width numeric fields within text. Other applications can undoubtedly be found.

For an explanation of the algorithm, it's simplest to look at these examples:

1.1 < 1.2

1.1a < 1.2

1.1 < 1.1.1

1.1 < 1.1a

1.1.a < 1.1a

1 < a

a < b

1 < 2

1.1-3 < 1.1-4

1.1-5 < 1.1.6

More precisely (but less comprehensibly), the two strings are treated as subunits delimited by periods or hyphens. Each subunit can contain any number of groups of digits or non-digits. If digit groups are being compared on both sides, a numeric comparison is used, otherwise a ASCII ordering is used. A group or subgroup with more units will win if all comparisons are equal. A period binds digit groups together more tightly than a hyphen.

Some packages use a different style of version numbering: a simple real number written as a decimal. Sort::Versions has limited support for this style: when comparing two subunits which are both digit groups, if either subunit has a leading zero, then both are treated like digits after a decimal point. So for example:

0002 < 1

1.06 < 1.5

This won't always work, because there won't always be a leading zero in real-number style version numbers. There is no way for Sort::Versions to know which style was intended. But a lot of the time it will do the right thing. If you are making up version numbers, the style with (possibly) more than one dot is the style to use.

USAGE

The function "versioncmp()" takes two arguments and compares them like "cmp". With perl 5.6 or later, you can also use this function directly in sorting:

```
@l = sort versioncmp qw(1.1 1.2 1.0.3);
```

The function "versions()" can be used directly as a sort function even on perl 5.005 and earlier, but its use is deprecated.

SEE ALSO

version, CPAN::Version which is part of the CPAN distribution.

REPOSITORY

<https://github.com/neilb/Sort-Versions>

AUTHOR

Ed Avis <ed@membled.com> and Matt Johnson <mwj99@doc.ic.ac.uk> for recent releases; the original author is Kenneth J. Albanowski <kjahds@kjahds.com>. Thanks to Hack Kampbjorn and Slaven Rezic for patches and bug reports.

COPYRIGHT AND LICENSE