

Package: semver (via r-universe)

September 4, 2024

Type Package

Title 'Semantic Versioning V2.0.0' Parser

Version 0.2.0

Description Tools and functions for parsing, rendering and operating on semantic version strings. Semantic versioning is a simple set of rules and requirements that dictate how version numbers are assigned and incremented as outlined at <http://semver.org>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports Rcpp (>= 0.12.8), assertthat, utils

LinkingTo Rcpp

URL <https://github.com/johndharrison/semver>

URLNote <https://github.com/johndharrison/semver>

BugReports <https://github.com/johndharrison/semver/issues>

Suggests testthat, covr, knitr, rmarkdown

RoxygenNote 5.0.1

SystemRequirements C++11

VignetteBuilder knitr

Repository <https://johndharrison.r-universe.dev>

RemoteUrl <https://github.com/johndharrison/semver>

RemoteRef HEAD

RemoteSha ee10af6418133f8f625df00477915e05f0272a3d

Contents

increment_version	2
parse_version	3
render_version	4
reset_version	5
semver	6
set_version	6
Index	8

increment_version	<i>Increment version field</i>
-------------------	--------------------------------

Description

Increment the version field. "Major", "minor" and "patch" fields are set as integers. Fields of lower precedence are reset to default value (0L)

Usage

```
increment_version(x, field, value, ...)

## S3 method for class 'svptr'
increment_version(x, field = c("major", "minor", "patch"),
  value, ...)

## S3 method for class 'svlist'
increment_version(x, field, value, ...)
```

Arguments

x	An object of class "svlist" or "svptr".
field	For the "svptr" class a character string detailing the field to set (major, minor etc...). For the "svlist" class a character vector detailing the field to set for each list member.
value	The value to increment the field by. For fields "major", "minor" and "patch" this is an integer. For the "svptr" class value is a scalar. For the "svlist" class value is an integer vector detailing the value to increment by for each list member.
...	Additional arguments passed on to methods.

Value

returns an object of class svptr or svlist

Examples

```
semver <- parse_version(c("1.2.3", "1.2.4-alpha+bld1"))
increment_version(semver[[2]], "major", 1L)
increment_version(semver[[2]], "major", -1L)
increment_version(semver, c("minor", "patch"), c(3L, 9L))
increment_version(semver, c("minor", "patch"), c(-1L, -3L))
increment_version(semver, c("minor"), 1L)
```

parse_version	<i>Parse semantic version</i>
---------------	-------------------------------

Description

Parse a character vector to semantic versions

Usage

```
parse_version(version)
```

Arguments

version A character vector

Value

A list of "svptr" objects. The list itself is of class "svlist"

Examples

```
semver <- parse_version(c("1.2.3", "1.2.4", "1.0.0", "1.2.4-beta", "2.1.3-alpha", "1.9.4"))
max(semver)
ver124a <- semver[semver > "1.2.4-alpha"]
sort(ver124a)
# compare versions
semver[[1]] < semver[[2]]

# compare against a version
semver > semver[1]

# compare against a character string
semver > "1.7.2"

# sort versions
sort(semver)
order(semver)
rank(semver)

# get summary statistics
min(semver)
```

```

max(semver)
range(semver)

# coerce versions
as.character(semver)
as.data.frame(semver)

```

render_version	<i>Render semantic version</i>
----------------	--------------------------------

Description

Render a semantic version list or a semantic version pointer as an R list giving the major, minor and patch version as an integer and the prerelease and build version as a character

Usage

```

render_version(x, ...)

## S3 method for class 'svptr'
render_version(x, ...)

## S3 method for class 'svlist'
render_version(x, ...)

```

Arguments

x	An object of class "svlist" or "svptr".
...	Additional arguments passed on to methods.

Value

A list of semantic versions represented as integers (major, minor, patch) and characters (prerelease, build)

Examples

```

semver <- parse_version(c("1.2.3", "1.2.4", "1.0.0", "1.2.4-beta+2", "2.1.3-alpha", "1.9.4"))
render_version(semver[[4]])
render_version(semver)

```

reset_version	<i>Reset version field</i>
---------------	----------------------------

Description

Reset the version field. "Major", "minor" and "patch" fields are set as integers, "prerelease" and "build" are set with character strings. Fields of lower precedence are reset to default value (0L or "")

Usage

```
reset_version(x, field, value, ...)

## S3 method for class 'svptr'
reset_version(x, field = c("major", "minor", "patch",
  "prerelease", "build"), value, ...)

## S3 method for class 'svlist'
reset_version(x, field, value, ...)
```

Arguments

x	An object of class "svlist" or "svptr".
field	For the "svptr" class a character string detailing the field to set (major, minor etc...). For the "svlist" class a character vector detailing the field to set for each list member.
value	The value to set the field to. For fields "major", "minor" and "patch" this is an integer. For fields "prerelease" and "build" this is a character. For the "svptr" class value is a scalar. For the "svlist" class value is a list detailing the value to set for each list member.
...	Additional arguments passed on to methods.

Value

returns an object of class svptr or svlist

Examples

```
semver <- parse_version(c("1.2.3", "1.2.4-alpha+bld1"))
reset_version(semver[[2]], "major", 1L)
semver[[1]] <- reset_version(semver[[1]], "major", 2L)
reset_version(semver, c("minor", "prerelease"), list(3L, "beta"))
reset_version(semver, c("prerelease"), list("hello", "world"))
```

semver	semver <i>parse semantic versions.</i>
--------	---

Description

semver has two functions: [parse_version](#), [render_version](#),

Details

[parse_version](#) takes as input a character vector with elements following the specification at <http://semver.org/>.

Comparisons can be done on the "svlist" and "svptr" classes returned by [parse_version](#). [Ops](#) and [summary](#) methods are implemented for both classes.

set_version	<i>Set version field</i>
-------------	--------------------------

Description

Set the version field. "Major", "minor" and "patch" fields are set as integers, "prerelease" and "build" are set with character strings.

Usage

```
set_version(x, field, value, ...)

## S3 method for class 'svptr'
set_version(x, field = c("major", "minor", "patch",
  "prerelease", "build"), value, ...)

## S3 method for class 'svlist'
set_version(x, field, value, ...)
```

Arguments

x	An object of class "svlist" or "svptr".
field	For the "svptr" class a character string detailing the field to set (major, minor etc...). For the "svlist" class a character vector detailing the field to set for each list member.
value	The value to set the field to. For fields "major", "minor" and "patch" this is an integer. For fields "prerelease" and "build" this is a character. For the "svptr" class value is a scalar. For the "svlist" class value is a list detailing the value to set for each list member.
...	Additional arguments passed on to methods.

Value

returns an object of class `svptr` or `svlist`

Examples

```
semver <- parse_version(c("1.2.3", "1.2.4-alpha+bld1"))
semver[[1]] <- set_version(semver[[1]], "major", 2L)
set_version(semver, c("prerelease", "build"), list("hello"))
set_version(semver, c("prerelease"), list("hello", "world"))
```

Index

`increment_version`, [2](#)

`Ops`, [6](#)

`parse_version`, [3](#), [6](#)

`render_version`, [4](#), [6](#)

`reset_version`, [5](#)

`semver`, [6](#)

`semver-package (semver)`, [6](#)

`set_version`, [6](#)

`summary`, [6](#)