semver-utils

1.1.0

Generated by Doxygen 1.8.10

Sat May 14 2016 18:47:01

Contents

Chapter 1

Main Page

1.1 Introduction

`libsemver` is a C++ library with C bindings that provides the following functionality:

- Parsing a version number into an object.
- Comparing versions.
- Modifying versions object.

Versions are checked against *Semantic Versioning 2.0.0* (`http://semver.org/`).

1.2 Changelog

See the History page.

1.3 Available Bindings

`libsemver` is a C++ library with C bindings which makes it available to a wide range of programming languages. If a programming language has C bindings, then `libsemver` can be used from it. The C binding provides all the functionality provided by the C++ implementation and it can be used as a fallback solution when the C++ API cannot be used.

1.4 libtool's versioning scheme

`libtool`'s versioning scheme is described by three integers: `current:revision:age` where:

- `current` is the most recent interface number implemented by the library.
- `revision` is the implementation number of the current interface.
- `age` is the difference between the newest and the oldest interface that the library implements.

1.5 Reporting Bugs and Suggestions

If you find problems or have suggestions about this program or this manual, please report them as new issues in the official GitHub repository at https://github.com/emcrisostomo/semver-utils. Please, read the CONTRIBUTING.md file for detailed instructions on how to contribute to fswatch.

Chapter 2

History

2.1 1:0:0

- Initial release.

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 5

File Index

5.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 6

Namespace Documentation

6.1 semver Namespace Reference

Main namespace of `libsemver`.

Classes

- class version

  Class that represents a version number.

Functions

- static std::vector< unsigned int > **parse_version** (const std::string &v)
- static void **match_prerelease** (const std::string &s)
- static void **match_metadata** (const std::string &s)
- static void **check_identifier** (const std::string &s)
- static const std::string **PRERELEASE_PATTERN** ("([0-9A-Za-z-]+(\\.[0-9A-Za-z-]+)*)")
- static const std::string **METADATA_PATTERN** ("([0-9A-Za-z-]+(\\.[0-9A-Za-z-]+)*)")

6.1.1 Detailed Description

Main namespace of `libsemver`.

Chapter 7

Class Documentation

7.1 semver_t Struct Reference

Public Attributes

- void * **ptr**

The documentation for this struct was generated from the following file:

- libsemver/c/libsemver.h

7.2 semver::version Class Reference

Class that represents a version number.

`#include <version.hpp>`

Public Member Functions

- version (const std::vector< unsigned int > versions, const std::string prerelease="", const std::string metadata="")

  Constructs a semver::version instance with the specified parameters. The parameters must comply with Semantic Versioning 2.0.0.
- std::string str () const

  Converts a version to its string representation.
- std::vector< unsigned int > get_version () const

  Gets the version components.
- unsigned int get_version (unsigned int index) const

  Gets the specified version component.
- std::string get_prerelease () const

  Gets the prerelease string.
- std::string get_metadata () const

  Gets the metadata string.
- version bump_major () const

  Bumps the major version component.
- version bump_minor () const

  Bumps the minor version component.

- version bump_patch () const

  Bumps the patch version component.
- version bump (unsigned int index) const

  Bumps the specified version component.
- version strip_prerelease () const

  Strips the prerelease component.
- version strip_metadata () const

  Strips the metadata component.
- bool is_release () const

  Checks whether the instance is a release version.
- bool operator== (const version &rh) const

  Checks two instances for equality.
- bool operator< (const version &rh) const

  Compares two instances.
- bool operator> (const version &rh) const

  Compares two instances.

Static Public Member Functions

- static version from_string (std::string v)

  Constructs a semver::version instance from a std::string.

7.2.1 Detailed Description

Class that represents a version number.

This class represents a version number complying with _Semantic Versioning 2.0.0 (http://semver.org/). As a supported extension this class allows version numbers to contain any number of components greater than 2.

Instances of this class are designed to be immutable.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 semver::version::version (const std::vector< unsigned int > *versions*, const std::string *prerelease* = "", const std::string *metadata* = "")

Constructs a semver::version instance with the specified parameters. The parameters must comply with *Semantic Versioning 2.0.0*.

Parameters

versions	The version components. As a supported extension this class allows versions numbers to contain any number of components greater than 2.
prerelease	An optional prerelease string.
metadata	An optional metadata string.

Returns

A semver::version instance.

Exceptions

std::invalid_argument	if the parameters do not comply with *Semantic Versioning 2.0.0*.

7.2.3 Member Function Documentation

7.2.3.1 version semver::version::bump (unsigned int *index*) const

Bumps the specified version component.

The component `index` is bumped and all components whose index is grater than `index` are set to 0. If `index` is greater than the current size `s` of the version number, then `index + 1 - s` components are added and set to 0.

Returns

A semver::version object representing the bumped version.

7.2.3.2 version semver::version::bump_major ( ) const

Bumps the major version component.

This method uses ::bump();

Returns

A semver::version object containing the bumped version.

7.2.3.3 version semver::version::bump_minor ( ) const

Bumps the minor version component.

This method uses ::bump();

Returns

A semver::version object representing the bumped version.

7.2.3.4 version semver::version::bump_patch ( ) const

Bumps the patch version component.

This method uses ::bump();

Returns

A semver::version object representing the bumped version.

7.2.3.5 version semver::version::from_string (std::string *v*) [static]

Constructs a semver::version instance from a std::string.

The version number `v` must comply with *Semantic Versioning 2.0.0*. As a supported extension this class allows version numbers to contain any number of components greater than 2.

Parameters

v	The version number to parse.

Returns

A semver::version instance.

Exceptions

std::invalid_argument	if `v` is not a valid version number.

7.2.3.6 std::string semver::version::get_metadata ( ) const

Gets the metadata string.

Returns

The metadata string.

7.2.3.7 std::string semver::version::get_prerelease ( ) const

Gets the prerelease string.

Returns

The prerelease string.

7.2.3.8 std::vector< unsigned int > semver::version::get_version ( ) const

Gets the version components.

Returns

The version components.

7.2.3.9 unsigned int semver::version::get_version (unsigned int *index*) const

Gets the specified version component.

Returns

The specified version component, or 0 if the specified `index` does not exist.

7.2.3.10 bool semver::version::is_release ( ) const

Checks whether the instance is a release version.

Returns

`true` if the instance represents a release version, `false` otherwise.

7.2.3.11 bool semver::version::operator< (const version & *rh*) const

Compares two instances.

An instance is less than another if its version number is less than the other's. Components are compared one by one, starting from the first. If the two components are equal, then the following component is considered. If the first component is less than the second, then `true` is returned, otherwise `false` is returned.

Parameters

v	The instance to be compared with.

Returns

`true` if this instance is less than `rh`, `false` otherwise.

7.2.3.12 bool semver::version::operator== (const version & *rh*) const

Checks two instances for equality.

Two instances are equal if and only if all its components are equal.

Parameters

v	The instance to be compared with.

Returns

`true` if `rh` is equal to this instance, `false` otherwise.

7.2.3.13 bool semver::version::operator> (const version & *rh*) const

Compares two instances.

An instance is greater than another if its version number is less than the other's. Components are compared one by one, starting from the first. If the two components are equal, then the following component is considered. If the first component is greater than the second, then `true` is returned, otherwise `false` is returned.

Parameters

v	The instance to be compared with.

Returns

`true` if this instance is greater than `rh`, `false` otherwise.

7.2.3.14 std::string semver::version::str ( ) const

Converts a version to its string representation.

Returns

The string representation of this instance.

7.2.3.15 version semver::version::strip_metadata ( ) const

Strips the metadata component.

Returns

A semver::version object representing the modified version.

7.2.3.16 version semver::version::strip_prerelease ( ) const

Strips the prerelease component.

Returns

A semver::version object representing the modified version.

The documentation for this class was generated from the following files:

- libsemver/c++/version.hpp
- libsemver/c++/version.cpp

Chapter 8

File Documentation

8.1 libsemver/c++/version.hpp File Reference

Header of the semver::version class.

```
#include <vector>
#include <string>
```

Classes

- class semver::version

  Class that represents a version number.

Namespaces

- semver

  Main namespace of `libsemver`.

8.1.1 Detailed Description

Header of the semver::version class.

This header defines the semver::version class, the central type of the C++ API of `libsemver` library.

Copyright

Copyright (c) 2016 Enrico M. Crisostomo

License:

GNU General Public License v. 3.0

Author

Enrico M. Crisostomo

Version

1.0.0

8.2 libsemver/c/libsemver.cpp File Reference

Main `libsemver` source file.

```
#include "libsemver.h"
#include "errors.h"
#include "../c++/version.hpp"
#include "../gettext_defs.h"
#include <exception>
#include <cstdlib>
```

Functions

- static void **semver_set_last_error** (int err)
- static void **semver_reset_last_error** ()
- int **semver_last_error** ()
- semver_t * **semver_from_string** (const char *str)
- semver_t * **semver_create** (const unsigned int *c_vers, const unsigned long c_vers_num, const char *prerelease, const char *metadata)
- void **semver_free** (semver_t *ver)
- const char * **semver_str** (semver_t *ver)
- unsigned int * **semver_get_versions** (semver_t *ver)
- unsigned int **semver_get_version** (semver_t *ver, unsigned int index)
- const char * **semver_get_prerelease** (semver_t *ver)
- const char * **semver_get_metadata** (semver_t *ver)
- semver_t * **semver_bump_major** (semver_t *ver)
- semver_t * **semver_bump_minor** (semver_t *ver)
- semver_t * **semver_bump_patch** (semver_t *ver)
- semver_t * **semver_bump** (semver_t *ver, unsigned int index)
- semver_t * **semver_strip_prerelease** (semver_t *ver)
- semver_t * **semver_strip_metadata** (semver_t *ver)
- bool **semver_is_release** (semver_t *ver)
- bool **semver_equals** (semver_t *lh, semver_t *rh)
- bool **semver_is_less** (semver_t *lh, semver_t *rh)
- bool **semver_is_greater** (semver_t *lh, semver_t *rh)

Variables

- static THREAD_LOCAL int **last_error**

8.2.1 Detailed Description

Main `libsemver` source file.

Copyright

Copyright (c) 2016 Enrico M. Crisostomo

License:

GNU General Public License v. 3.0

Author

Enrico M. Crisostomo

Version

1.0.0

8.3 libsemver/c/libsemver.h File Reference

Header of the `libsemver` library.

```
#include <stdbool.h>
```

Classes

- struct semver_t

Typedefs

- typedef struct semver_t **semver_t**

Functions

- int **semver_last_error** ()
- semver_t * **semver_from_string** (const char *str)
- semver_t * **semver_create** (const unsigned int *c_vers, const unsigned long c_vers_num, const char *prerelease, const char *metadata)
- void **semver_free** (semver_t *ver)
- const char * **semver_str** (semver_t *ver)
- unsigned int * **semver_get_versions** (semver_t *ver)
- unsigned int **semver_get_version** (semver_t *ver, unsigned int index)
- const char * **semver_get_prerelease** (semver_t *ver)
- const char * **semver_get_metadata** (semver_t *ver)
- semver_t * **semver_bump_major** (semver_t *ver)
- semver_t * **semver_bump_minor** (semver_t *ver)
- semver_t * **semver_bump_patch** (semver_t *ver)
- semver_t * **semver_bump** (semver_t *ver, unsigned int index)
- semver_t * **semver_strip_prerelease** (semver_t *ver)
- semver_t * **semver_strip_metadata** (semver_t *ver)
- bool **semver_is_release** (semver_t *ver)
- bool **semver_equals** (semver_t *lh, semver_t *rh)
- bool **semver_is_less** (semver_t *lh, semver_t *rh)
- bool **semver_is_greater** (semver_t *lh, semver_t *rh)

8.3.1 Detailed Description

Header of the `libsemver` library.

This header file defines the API of the `libsemver` library.

Copyright

License:

Author

Enrico M. Crisostomo

Version

1.0.0

Index