# CFStringComparators

Generated by Doxygen 1.8.11

# Contents


**1 License** **1**

**2 Namespace Index** **3**

2.1 Namespace List . . . . . 3

**3 Class Index** **5**

3.1 Class List . . . . . 5

**4 File Index** **7**

4.1 File List . . . . . 7

**5 Namespace Documentation** **9**

5.1 CaseInsensitive Namespace Reference . . . . . 9

5.1.1 Detailed Description . . . . . 9

5.2 CFStringComparators Namespace Reference . . . . . 9

5.2.1 Detailed Description . . . . . 9

**6 Class Documentation** **11**

6.1 CFStringComparators::CaseInsensitive::Equality Class Reference . . . . . 11

6.1.1 Detailed Description . . . . . 11

6.1.2 Member Function Documentation . . . . . 11

6.1.2.1 operator()(const std::string &lhs, const std::string &rhs) const . . . . . 11

6.2 CFStringComparators::CaseInsensitive::Sort Class Reference . . . . . 12

6.2.1 Detailed Description . . . . . 12

6.2.2 Member Function Documentation . . . . . 12

6.2.2.1 operator()(const std::string &lhs, const std::string &rhs) const . . . . . 12

**7 File Documentation** **13**

7.1 CaseInsensitive.hpp File Reference . . . . . 13

7.1.1 Detailed Description . . . . . 14

7.2 CaseInsensitive/Equality.cpp File Reference . . . . . 14

7.2.1 Detailed Description . . . . . 15

7.3 CaseInsensitive/Equality.hpp File Reference . . . . . 15

7.3.1 Detailed Description . . . . . 16

7.4 CaseInsensitive/Sort.cpp File Reference . . . . . 16

7.4.1 Detailed Description . . . . . 17

7.5 CaseInsensitive/Sort.hpp File Reference . . . . . 17

7.5.1 Detailed Description . . . . . 19

7.6 CFStringComparators.hpp File Reference . . . . . 19

7.6.1 Detailed Description . . . . . 19

**Index** **21**

# Chapter 1

# License

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see http://www.gnu.org/licenses/.

# Chapter 2

# Namespace Index

## 2.1 Namespace List

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


CaseInsensitive
: `CaseInsensitive` is a part of the `CFStringComparators` C++ comparator expansion pack that provides case insensitive `std::string` comparators for use with standard C++ data structures . . . . . 9

CFStringComparators
: `CFStringComparators` is a C++ comparator expansion pack that provides extra `std::string` comparators for use with standard C++ data structures (`std::map`, `std::set`, etc) . . . . . 9

# Chapter 3

# Class Index

## 3.1 Class List

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


CFStringComparators::CaseInsensitive::Equality
: Function object used to compare two `std::string` objects in order to determine equality of their contents . . . . . 11

CFStringComparators::CaseInsensitive::Sort
: Function object used to compare two `std::string` objects in order to determine sort order 12

# Chapter 4

# File Index

## 4.1 File List

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


CaseInsensitive.hpp . . . . . 13
CFStringComparators.hpp . . . . . 19
CaseInsensitive/Equality.cpp . . . . . 14
CaseInsensitive/Equality.hpp . . . . . 15
CaseInsensitive/Sort.cpp . . . . . 16
CaseInsensitive/Sort.hpp . . . . . 17

# Chapter 5

# Namespace Documentation

## 5.1 CaseInsensitive Namespace Reference

`CaseInsensitive` is a part of the `CFStringComparators` C++ comparator expansion pack that provides case insensitive `std::string` comparators for use with standard C++ data structures.

### 5.1.1 Detailed Description

`CaseInsensitive` is a part of the `CFStringComparators` C++ comparator expansion pack that provides case insensitive `std::string` comparators for use with standard C++ data structures.

## 5.2 CFStringComparators Namespace Reference

`CFStringComparators` is a C++ comparator expansion pack that provides extra `std::string` comparators for use with standard C++ data structures (`std::map`, `std::set`, etc).

### 5.2.1 Detailed Description

`CFStringComparators` is a C++ comparator expansion pack that provides extra `std::string` comparators for use with standard C++ data structures (`std::map`, `std::set`, etc).

# Chapter 6

# Class Documentation

## 6.1 CFStringComparators::CaseInsensitive::Equality Class Reference

Function object used to compare two `std::string` objects in order to determine equality of their contents.

`#include <Equality.hpp>`

### Public Member Functions

- bool operator() (const std::string &lhs, const std::string &rhs) const
  *Compares two strings in a case-insensitive manner to determine equality.*

### 6.1.1 Detailed Description

Function object used to compare two `std::string` objects in order to determine equality of their contents.

This object implements the function call operator to provide a callable class interface for string comparison.

**See also**

`Equality::operator()()` for details regarding actual string comparison.

### 6.1.2 Member Function Documentation

#### 6.1.2.1 bool CFStringComparators::CaseInsensitive::Equality::operator() ( const std::string & *lhs*, const std::string & *rhs* ) const

Compares two strings in a case-insensitive manner to determine equality.

By making use of the `strncasecmp()` POSIX function, one can define a custom comparator for strings that is case-insensitive and memory-safe

This method for comparing strings is *probably not* Unicode compatible

**See also**

`strncasecmp()` for more details on string comparison

**Returns**

`true` if both strings are equal, `false` otherwise

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

- CaseInsensitive/Equality.hpp
- CaseInsensitive/Equality.cpp

## 6.2 CFStringComparators::CaseInsensitive::Sort Class Reference

Function object used to compare two `std::string` objects in order to determine sort order.

`#include <Sort.hpp>`

### Public Member Functions

- bool operator() (const std::string &lhs, const std::string &rhs) const
  *Compares two strings in a case-insensitive manner to determine sort order.*

### 6.2.1 Detailed Description

Function object used to compare two `std::string` objects in order to determine sort order.

This object implements the function call operator to provide a callable class interface for string comparison.

**See also**

`Sort::operator()()` for details regarding actual string comparison.

### 6.2.2 Member Function Documentation

#### 6.2.2.1 bool CFStringComparators::CaseInsensitive::Sort::operator() ( const std::string & *lhs*, const std::string & *rhs* ) const

Compares two strings in a case-insensitive manner to determine sort order.

By making use of the `strncasecmp()` POSIX function, one can define a custom comparator for strings that is case-insensitive and memory-safe (only measuring up to the smallest string's length)

This method for comparing strings is *probably not* Unicode compatible

**See also**

`strncasecmp()` for more details on string comparison

**Returns**

`true` if the left string sorts before the other, `false` otherwise

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

- CaseInsensitive/Sort.hpp
- CaseInsensitive/Sort.cpp

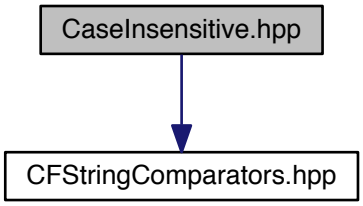
# Chapter 7

# File Documentation

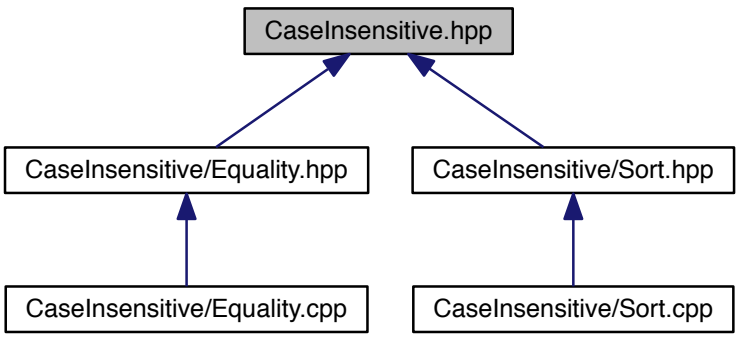
## 7.1 CaseInsensitive.hpp File Reference

```
#include "CFStringComparators.hpp"
```

Include dependency graph for CaseInsensitive.hpp:




This graph shows which files directly or indirectly include this file:



### Namespaces

- CFStringComparators

  *CFStringComparators is a C++ comparator expansion pack that provides extra std::string comparators for use with standard C++ data structures (std::map, std::set, etc).*
- CaseInsensitive

  *CaseInsensitive is a part of the CFStringComparators C++ comparator expansion pack that provides case insensitive std::string comparators for use with standard C++ data structures.*

### 7.1.1 Detailed Description

**Copyright**

Copyright 2016 Clay Freeman. All rights reserved

**License:**

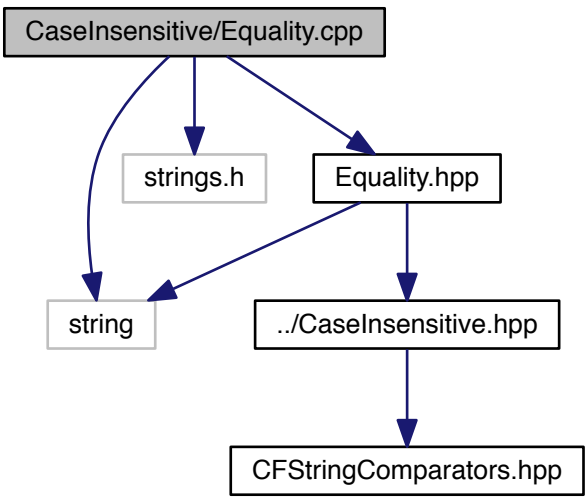
GNU Lesser General Public License v3 (LGPL-3.0)

Forward declaration of the CaseInsensitive namespace and related items.

## 7.2 CaseInsensitive/Equality.cpp File Reference

```
#include <string>
#include <strings.h>
#include "Equality.hpp"
```

Include dependency graph for Equality.cpp:



### Namespaces

- CFStringComparators

  *CFStringComparators is a C++ comparator expansion pack that provides extra std::string comparators for use with standard C++ data structures (std::map, std::set, etc).*

### 7.2.1 Detailed Description

#### Copyright

Copyright 2016 Clay Freeman. All rights reserved

#### License:

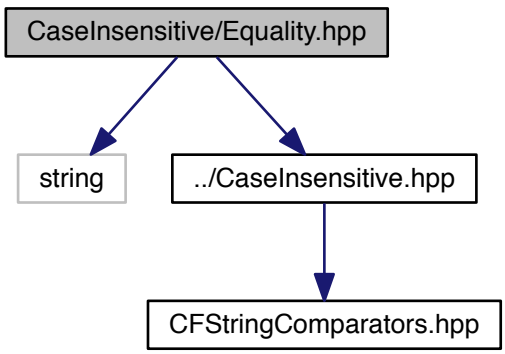
GNU Lesser General Public License v3 (LGPL-3.0)

Implementation source for the `Equality` object.

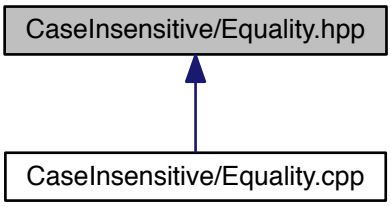
## 7.3 CaseInsensitive/Equality.hpp File Reference

```
#include <string>
#include "../CaseInsensitive.hpp"
```

Include dependency graph for Equality.hpp:



This graph shows which files directly or indirectly include this file:




### Classes

- class CFStringComparators::CaseInsensitive::Equality

  *Function object used to compare two `std::string` objects in order to determine equality of their contents.*

### Namespaces

- CFStringComparators

  *`CFStringComparators` is a C++ comparator expansion pack that provides extra `std::string` comparators for use with standard C++ data structures (`std::map`, `std::set`, etc).*

### 7.3.1 Detailed Description

#### Copyright

Copyright 2016 Clay Freeman. All rights reserved

#### License:

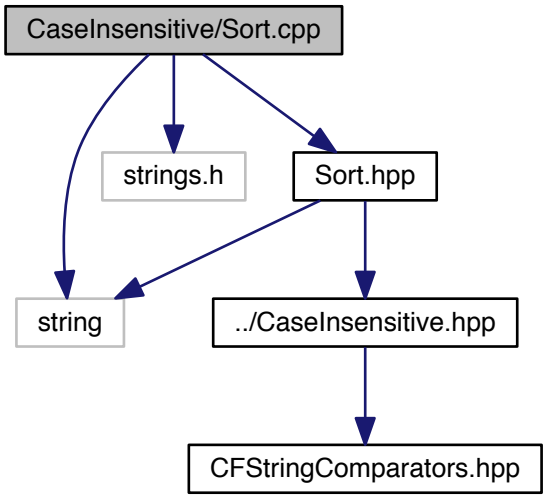
GNU Lesser General Public License v3 (LGPL-3.0)

Implementation reference for the `Equality` object.

## 7.4 CaseInsensitive/Sort.cpp File Reference

```
#include <string>
#include <strings.h>
#include "Sort.hpp"
```

Include dependency graph for Sort.cpp:




### Namespaces

- CFStringComparators

  *`CFStringComparators` is a C++ comparator expansion pack that provides extra `std::string` comparators for use with standard C++ data structures (`std::map`, `std::set`, etc).*

### 7.4.1 Detailed Description

**Copyright**

Copyright 2016 Clay Freeman. All rights reserved

**License:**

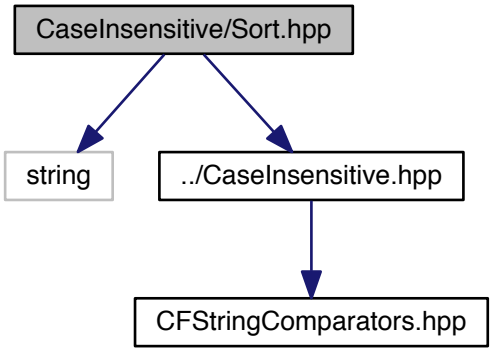
GNU Lesser General Public License v3 (LGPL-3.0)

Implementation source for the `Sort` object.

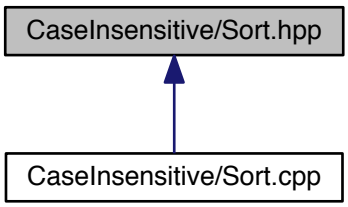
## 7.5 CaseInsensitive/Sort.hpp File Reference

```
#include <string>
#include "../CaseInsensitive.hpp"
```

Include dependency graph for Sort.hpp:




This graph shows which files directly or indirectly include this file:




## Classes

- class CFStringComparators::CaseInsensitive::Sort

  *Function object used to compare two `std::string` objects in order to determine sort order.*

## Namespaces

- CFStringComparators

  *`CFStringComparators` is a C++ comparator expansion pack that provides extra `std::string` comparators for use with standard C++ data structures (`std::map`, `std::set`, etc).*

### 7.5.1 Detailed Description

**Copyright**

Copyright 2016 Clay Freeman. All rights reserved

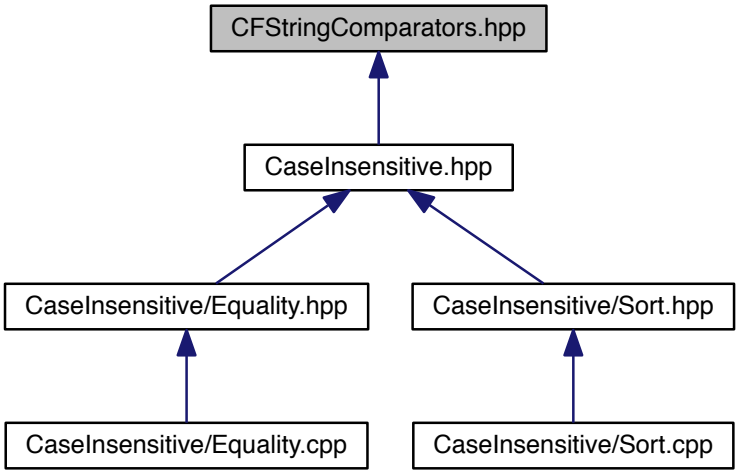
**License:**

GNU Lesser General Public License v3 (LGPL-3.0)

Implementation reference for the `Sort` object.

## 7.6 CFStringComparators.hpp File Reference

This graph shows which files directly or indirectly include this file:




### Namespaces

- CFStringComparators

  *`CFStringComparators` is a C++ comparator expansion pack that provides extra `std::string` comparators for use with standard C++ data structures (`std::map`, `std::set`, etc).*

### 7.6.1 Detailed Description

**Copyright**

Copyright 2016 Clay Freeman. All rights reserved

**License:**

GNU Lesser General Public License v3 (LGPL-3.0)

Forward declaration of the `CFStringComparators` namespace and related items.

# Index

CFStringComparators, 9
CFStringComparators.hpp, 19
CFStringComparators::CaseInsensitive::Equality, 11
    operator(), 11
CFStringComparators::CaseInsensitive::Sort, 12
    operator(), 12
CaseInsensitive, 9
CaseInsensitive.hpp, 13
CaseInsensitive/Equality.cpp, 14
CaseInsensitive/Equality.hpp, 15
CaseInsensitive/Sort.cpp, 16
CaseInsensitive/Sort.hpp, 17

operator()
    CFStringComparators::CaseInsensitive::Equality, 11
    CFStringComparators::CaseInsensitive::Sort, 12