Shared String View

0.1

Generated by Doxygen 1.9.1

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 std::hash< Ghoti::shared_string_view > Struct Reference	5
3.1.1 Detailed Description	5
3.2 Ghoti::shared_string_view Class Reference	6
3.2.1 Constructor & Destructor Documentation	7
3.2.1.1 shared_string_view() [1/4]	7
3.2.1.2 shared_string_view() [2/4]	7
3.2.1.3 shared_string_view() [3/4]	8
3.2.1.4 shared_string_view() [4/4]	8
3.2.2 Member Function Documentation	8
3.2.2.1 begin()	8
3.2.2.2 end()	8
3.2.2.3 length()	9
3.2.2.4 operator std::string_view()	9
3.2.2.5 operator+()	9
3.2.2.6 operator+=()	9
3.2.2.7 operator<=>()	10
3.2.2.8 operator==()	10
3.2.2.9 operator[]()	11
3.2.2.10 rbegin()	11
3.2.2.11 rend()	11
3.2.2.12 substr()	11
	40
4 File Documentation	13
4.1 include/shared_string_view.hpp File Reference	13
4.1.1 Detailed Description	14
4.1.2 Function Documentation	14
4.1.2.1 operator<<<()	14
4.2 src/shared_string_view.cpp File Reference	15
4.2.1 Detailed Description	15
4.3 test/test.cpp File Reference	15
4.3.1 Detailed Description	16
Index	17

# **Class Index**

## 1.1 Class List

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

std::hash< Ghoti::shared_string_view >	
Hashing function, consistent with std::string_view	5
Ghoti::shared_string_view	6

2 Class Index

# File Index

## 2.1 File List

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

include/shared string view.hpp	
Header file containing the definitions of the shared_string_view class	13
src/shared_string_view.cpp	
Define the shared_string_view class	15
test/test.cpp	
Test the shared string view behavior	15

File Index

## **Class Documentation**

## 3.1 std::hash< Ghoti::shared\_string\_view > Struct Reference

Hashing function, consistent with std::string\_view.

#include <shared\_string\_view.hpp>

#### **Public Member Functions**

• std::size\_t operator() (const Ghoti::shared\_string\_view &ssv) const noexcept

## 3.1.1 Detailed Description

Hashing function, consistent with std::string\_view.

#### **Parameters**

ssv | The shared\_string\_view to be hashed.

#### Returns

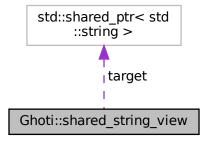
The hashed value.

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

- include/shared\_string\_view.hpp
- src/shared\_string\_view.cpp

## 3.2 Ghoti::shared string view Class Reference

Collaboration diagram for Ghoti::shared\_string\_view:



#### **Public Member Functions**

shared string view (const char \*s)

Constructor.

• shared\_string\_view (const char \*s, size\_t length)

Constructor.

• shared\_string\_view (const std::string &s)

Constructor.

operator std::string\_view () const

Provides a string\_view variant of the shared\_string\_view object.

• size\_t length () const

Return the length of the string represented by the view.

shared\_string\_view substr (size\_t offset, size\_t length)

Calculate a substring based on the current shared\_string\_view.

bool operator== (const shared\_string\_view &ssv) const

Compare two shared\_string\_view objects.

std::weak\_ordering operator<=> (const shared\_string\_view &ssv) const

Perform a three-way comparison on two shared\_string\_view objects.

shared\_string\_view & operator+= (const std::string &rhs)

Perform a concatenation of the supplied string to the existing string view object and apply it to the existing string view object.

shared string view operator+ (const std::string &rhs)

Perform a concatenation of the supplied string to the existing string view object.

• std::string\_view::const\_iterator begin () const

Provide an iterator from the beginning of the view.

• std::string\_view::const\_iterator end () const

Provide an iterator pointing to the end of the view.

• std::string\_view::const\_reverse\_iterator rbegin () const

Provide a reverse iterator from the end of the view.

• std::string\_view::const\_reverse\_iterator rend () const

Provide a reverse iterator pointing to the beginning of the view.

• char operator[] (size\_t pos) const

Perform an index operation into the target string.

#### **Private Member Functions**

shared\_string\_view ()

Private constructor.

#### **Private Attributes**

• std::shared\_ptr< std::string > target

The shared target string pointed to by this object.

• size\_t start

The offset into the shared target string at which this view begins.

size\_t len

The length of the view.

#### 3.2.1 Constructor & Destructor Documentation

#### 3.2.1.1 shared\_string\_view() [1/4]

## Constructor.

#### **Parameters**

s A C-string used to construct the string.

## 3.2.1.2 shared\_string\_view() [2/4]

#### Constructor.

#### **Parameters**

s	A C-string used to construct the string.
length	The length of the C-string.

#### 3.2.1.3 shared\_string\_view() [3/4]

```
\begin{tabular}{ll} shared\_string\_view::shared\_string\_view (\\ const std::string \& s ) \end{tabular}
```

Constructor.

**Parameters** 

s A String object used to construct the string.

#### 3.2.1.4 shared\_string\_view() [4/4]

```
shared_string_view::shared_string_view ( ) [private]
```

Private constructor.

This constructor is private because it will create an object whose target is not initialized, which should not be done in general.

#### 3.2.2 Member Function Documentation

#### 3.2.2.1 begin()

```
string_view::const_iterator shared_string_view::begin ( ) const
```

Provide an iterator from the beginning of the view.

Returns

A forward iterator.

#### 3.2.2.2 end()

```
\verb|string_view::const_iterator shared_string_view::end () const|
```

Provide an iterator pointing to the end of the view.

Returns

An ending iterator.

#### 3.2.2.3 length()

```
size\_t shared\_string\_view::length () const
```

Return the length of the string represented by the view.

The shared string may be longer, but this is the length of the substring that this view represents.

Returns

The length of the string represented by the view.

#### 3.2.2.4 operator std::string\_view()

```
Ghoti::shared_string_view::operator std::string_view ( ) const
```

Provides a string\_view variant of the <a href="mailto:shared\_string\_view">shared\_string\_view</a> object.

It is up to the programmer to ensure that the <a href="mailto:shared\_string\_view">shared\_string\_view</a> object remains in scope while the string\_view is in use.

#### 3.2.2.5 operator+()

Perform a concatenation of the supplied string to the existing string view object.

Return a new string view.

#### **Parameters**

```
rhs A string to be appended to the shared_string_view object.
```

#### Returns

The new shared\_string\_view resulting from the concatenation.

#### 3.2.2.6 operator+=()

Perform a concatenation of the supplied string to the existing string view object and apply it to the existing string view object.

If the target string can be appended to safely, then that will be done. Otherwise, a new internal string will be created.

Because this may modify the target string, all previously-provided std::string\_view references will be invalidated. This is similar to the behavior of std::string.cstr(), in which modifying the string will invalidate the c-string pointer.

#### **Parameters**

```
rhs A string to be appended to the shared_string_view object.
```

#### Returns

The amended shared\_string\_view resulting from the concatenation.

#### 3.2.2.7 operator<=>()

Perform a three-way comparison on two shared\_string\_view objects.

#### **Parameters**

```
ssv The right hand side operator.
```

#### Returns

A weak ordering indicator of the two objects.

#### 3.2.2.8 operator==()

Compare two shared\_string\_view objects.

#### **Parameters**

ssv The right hand side operator.

#### Returns

True if the objects have equivalent values, false otherwise.

#### 3.2.2.9 operator[]()

Perform an index operation into the target string.

#### **Parameters**

```
pos The 0-based index position.
```

#### Returns

The character at the position requested.

#### 3.2.2.10 rbegin()

```
string_view::const_reverse_iterator shared_string_view::rbegin ( ) const
```

Provide a reverse iterator from the end of the view.

#### Returns

A reverse iterator.

#### 3.2.2.11 rend()

```
string_view::const_reverse_iterator shared_string_view::rend ( ) const
```

Provide a reverse iterator pointing to the beginning of the view.

#### Returns

An reverse ending iterator.

#### 3.2.2.12 substr()

Calculate a substring based on the current shared\_string\_view.

If the substring is out of range, then an empty view will be provided. If the substring length requested is greater than what is available, then the returned substring will contain as many characters as possible, within the limits of the parent string view range.

#### **Parameters**

offset	The 0-based offset from which the substring should start.
length	The length of the substring desired.

#### Returns

A new shared\_string\_view of the requested substring.

Here is the call graph for this function:



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

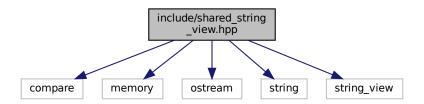
- include/shared\_string\_view.hpp
- src/shared\_string\_view.cpp

## **File Documentation**

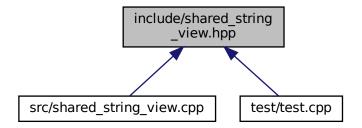
## 4.1 include/shared\_string\_view.hpp File Reference

Header file containing the definitions of the shared\_string\_view class.

```
#include <compare>
#include <memory>
#include <ostream>
#include <string>
#include <string_view>
Include dependency graph for shared_string_view.hpp:
```



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



14 File Documentation

#### **Classes**

- class Ghoti::shared\_string\_view
- struct std::hash< Ghoti::shared\_string\_view >

Hashing function, consistent with std::string\_view.

#### **Functions**

std::ostream & Ghoti::operator<< (std::ostream &out, const Ghoti::shared\_string\_view &ssv)</li>
 Insertion operator.

## 4.1.1 Detailed Description

Header file containing the definitions of the shared\_string\_view class.

## 4.1.2 Function Documentation

#### 4.1.2.1 operator << ()

Insertion operator.

#### **Parameters**

out	The output stream to be written to.
SSV	The shared_string_view to be inserted into the stream.

#### Returns

The output stream.

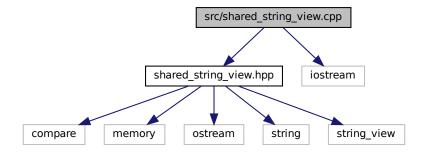
Here is the call graph for this function:



## 4.2 src/shared\_string\_view.cpp File Reference

Define the shared\_string\_view class.

```
#include "shared_string_view.hpp"
#include <iostream>
Include dependency graph for shared_string_view.cpp:
```



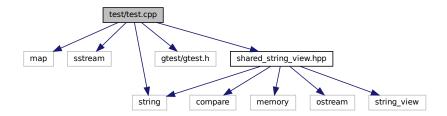
#### 4.2.1 Detailed Description

Define the shared\_string\_view class.

## 4.3 test/test.cpp File Reference

Test the shared\_string\_view behavior.

```
#include <map>
#include <sstream>
#include <string>
#include <gtest/gtest.h>
#include "shared_string_view.hpp"
Include dependency graph for test.cpp:
```



16 File Documentation

## **Functions**

- TEST (Constructor, Length)
- **TEST** (Operator, string\_stream)
- TEST (Operator, Extraction)
- TEST (Operator, ThreeWayComparison)
- TEST (Operator, PlusEqual)
- TEST (Operator, Plus)
- TEST (Method, substr)
- **TEST** (Method, ForwardIterator)
- TEST (Method, Reverselterator)
- TEST (Method, Index)
- TEST (Aux, Hash)
- int main (int argc, char \*\*argv)

## 4.3.1 Detailed Description

Test the shared\_string\_view behavior.

## Index

```
begin
    Ghoti::shared_string_view, 8
end
    Ghoti::shared_string_view, 8
Ghoti::shared_string_view, 6
    begin, 8
     end, 8
    length, 8
    operator std::string_view, 9
    operator<=>, 10
    operator+, 9
    operator+=, 9
    operator==, 10
     operator[], 10
     rbegin, 11
     rend, 11
     shared_string_view, 7, 8
     substr, 11
include/shared_string_view.hpp, 13
length
     Ghoti::shared_string_view, 8
operator std::string_view
    Ghoti::shared_string_view, 9
operator<<
     shared_string_view.hpp, 14
operator<=>
     Ghoti::shared_string_view, 10
operator+
     Ghoti::shared_string_view, 9
operator+=
     Ghoti::shared_string_view, 9
operator==
    Ghoti::shared_string_view, 10
operator[]
    Ghoti::shared_string_view, 10
rbegin
     Ghoti::shared string view, 11
rend
    Ghoti::shared_string_view, 11
shared_string_view
     Ghoti::shared_string_view, 7, 8
shared_string_view.hpp
     operator<<, 14
src/shared_string_view.cpp, 15
```

```
std::hash< Ghoti::shared_string_view >, 5
substr
     Ghoti::shared_string_view, 11
test/test.cpp, 15
```