

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 Ghoti::shared_string_view Class Reference	5
3.1.1 Constructor & Destructor Documentation	6
3.1.1.1 shared_string_view() [1/3]	6
3.1.1.2 shared_string_view() [2/3]	6
3.1.1.3 shared_string_view() [3/3]	7
3.1.2 Member Function Documentation	7
3.1.2.1 begin()	7
3.1.2.2 end()	7
3.1.2.3 length()	8
3.1.2.4 operator std::string_view()	8
3.1.2.5 operator<=>()	8
3.1.2.6 operator==()	8
3.1.2.7 operator[]()	9
3.1.2.8 rbegin()	9
3.1.2.9 rend()	9
3.1.2.10 substr()	10
4 File Documentation	11
4.1 include/shared_string_view.hpp File Reference	11
4.1.1 Detailed Description	12
4.2 src/shared_string_view.cpp File Reference	12
4.2.1 Detailed Description	12
4.3 test/test.cpp File Reference	13
4.3.1 Detailed Description	13
Index	15

Chapter 1

Class Index

1.1 Class List

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

Ghoti::shared_string_view	5
---	---

Chapter 2

File Index

2.1 File List

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

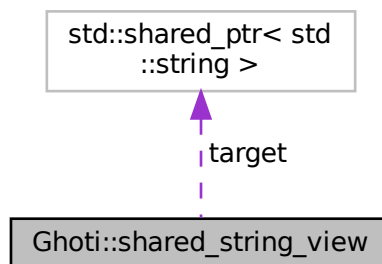
include/ shared_string_view.hpp	
Header file containing the definitions of the shared_string_view class	11
src/ shared_string_view.cpp	
Define the shared_string_view class	12
test/ test.cpp	
Test the shared_string_view behavior	13

Chapter 3

Class Documentation

3.1 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.

- `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 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.

3.1.1 Constructor & Destructor Documentation

3.1.1.1 shared_string_view() [1/3]

```
shared_string_view::shared_string_view (
    const char * s )
```

Constructor.

Parameters

<code>s</code>	A C-string used to construct the string.
----------------	--

3.1.1.2 shared_string_view() [2/3]

```
shared_string_view::shared_string_view (
    const char * s,
    size_t length )
```

Constructor.

Parameters

<i>s</i>	A C-string used to construct the string.
<i>length</i>	The length of the C-string.

3.1.1.3 shared_string_view() [3/3]

```
shared_string_view::shared_string_view (
    const std::string & s )
```

Constructor.

Parameters

<i>s</i>	A String object used to construct the string.
----------	---

3.1.2 Member Function Documentation**3.1.2.1 begin()**

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

Provide an iterator from the beginning of the view.

Returns

A forward iterator.

3.1.2.2 end()

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

Provide an iterator pointing to the end of the view.

Returns

An ending iterator.

3.1.2.3 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.

3.1.2.4 operator std::string_view()

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

Provides a `string_view` variant of the `shared_string_view` object.

It is up to the programmer to ensure that the `shared_string_view` object remains in scope while the `string_view` is in use.

3.1.2.5 operator<=>()

```
std::weak_ordering shared_string_view::operator<=> (
    const shared_string_view & ssv ) const
```

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.

3.1.2.6 operator==()

```
bool shared_string_view::operator==(
    const shared_string_view & ssv ) const
```

Compare two `shared_string_view` objects.

Parameters

<i>ssv</i>	The right hand side operator.
------------	-------------------------------

Returns

True if the objects have equivalent values, false otherwise.

3.1.2.7 operator[]()

```
char shared_string_view::operator[] (
    size_t pos ) const
```

Perform an index operation into the target string.

Parameters

<i>pos</i>	The 0-based index position.
------------	-----------------------------

Returns

The character at the position requested.

3.1.2.8 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.

3.1.2.9 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.

3.1.2.10 substr()

```
shared_string_view shared_string_view::substr (
    size_t offset,
    size_t length )
```

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.

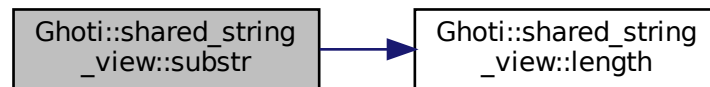
Parameters

<i>offset</i>	The 0-based offset from which the substring should start.
<i>length</i>	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](#)

Chapter 4

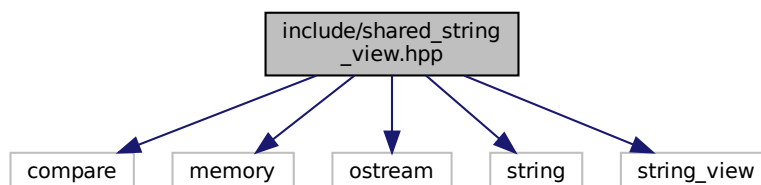
File Documentation

4.1 include/shared_string_view.hpp File Reference

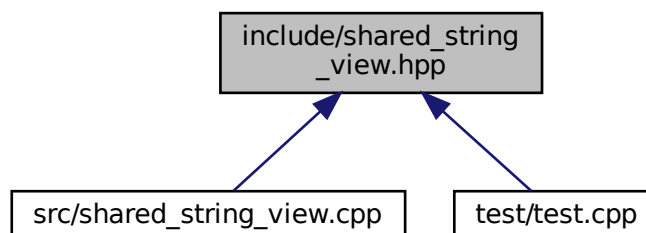
Header file containing the definitions of the shared_string_view class.

```
#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:



Classes

- class [Ghoti::shared_string_view](#)

Functions

- `std::ostream & Ghoti::operator<< (std::ostream &out, const Ghoti::shared_string_view &ssv)`

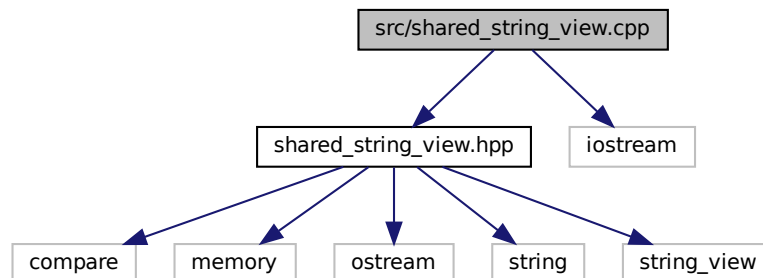
4.1.1 Detailed Description

Header file containing the definitions of the `shared_string_view` class.

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:
```



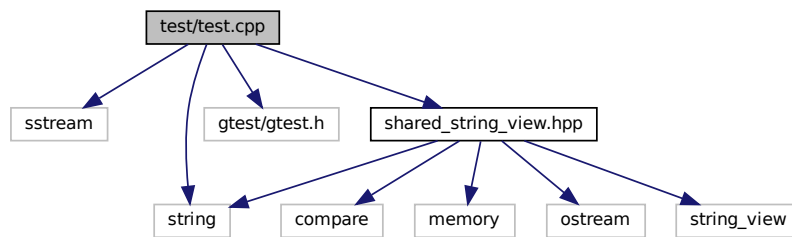
4.2.1 Detailed Description

Define the `shared_string_view` class.

4.3 test/test.cpp File Reference

Test the `shared_string_view` behavior.

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



Functions

- **TEST** (Constructor, Length)
- **TEST** (Operator, `string_stream`)
- **TEST** (Operator, Extraction)
- **TEST** (Operator, `ThreeWayComparison`)
- **TEST** (Method, `substr`)
- **TEST** (Method, `ForwardIterator`)
- **TEST** (Method, `ReverseIterator`)
- **TEST** (Method, `Index`)
- `int main` (`int argc`, `char **argv`)

4.3.1 Detailed Description

Test the `shared_string_view` behavior.

Index

- begin
 - Ghoti::shared_string_view, [7](#)
- end
 - Ghoti::shared_string_view, [7](#)
- Ghoti::shared_string_view, [5](#)
 - begin, [7](#)
 - end, [7](#)
 - length, [7](#)
 - operator std::string_view, [8](#)
 - operator<=>, [8](#)
 - operator==, [8](#)
 - operator[], [9](#)
 - rbegin, [9](#)
 - rend, [9](#)
 - shared_string_view, [6](#), [7](#)
 - substr, [9](#)
- include/shared_string_view.hpp, [11](#)
- length
 - Ghoti::shared_string_view, [7](#)
- operator std::string_view
 - Ghoti::shared_string_view, [8](#)
- operator<=>
 - Ghoti::shared_string_view, [8](#)
- operator==
 - Ghoti::shared_string_view, [8](#)
- operator[]
 - Ghoti::shared_string_view, [9](#)
- rbegin
 - Ghoti::shared_string_view, [9](#)
- rend
 - Ghoti::shared_string_view, [9](#)
- shared_string_view
 - Ghoti::shared_string_view, [6](#), [7](#)
- src/shared_string_view.cpp, [12](#)
- substr
 - Ghoti::shared_string_view, [9](#)
- test/test.cpp, [13](#)