«deque»

Generated by Doxygen 1.8.12

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

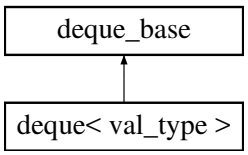
Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 deque< val_type > Class Template Reference

Inheritance diagram for deque< val_type >:



Public Types

- typedef node< val_type > **Node**
- typedef deque_iterator< val_type, val_type &, val_type * > **iterator**
- typedef deque_iterator< val_type, const val_type &, const val_type * > **const_iterator**

Public Member Functions

- **deque** (size_t n)
- deque (size_t n, const val_type &var)

  Fill Contsructor.
- deque (deque &&move)

  Move Constructor.
- deque (const deque &Obj)

  Copy Constructor.
- iterator **begin** ()
- const_iterator **begin** () const
- iterator **end** ()
- const_iterator **end** () const
- const_iterator **cend** () const
- const_iterator **cbegin** () const
- deque & **operator=** (const deque &Obj)
- deque & **operator=** (deque &&x)
- size_t **size** () const

- void **resize** (size_t n)
- bool **empty** ()
- val_type & **at** (size_t n)
- const val_type & **at** (size_t n) const
- val_type & **front** ()
- const val_type & **front** () const
- val_type & **back** ()
- const val_type & **back** () const
- void **push_front** (const val_type &val)
- void **push_back** (const val_type &val)
- void **pop_back** ()
- void **pop_front** ()
- void **clear** ()
- void **assign** (size_t n, const val_type &val)
- iterator **insert** (const_iterator position, const val_type &val)
- template<class InputIterator >
  iterator **insert** (const_iterator position, InputIterator first, InputIterator last)
- val_type & **operator[ ]** (size_t n)
- const val_type & **operator[ ]** (size_t n) const
- void **swap** (deque &b)
- iterator **erase** (const_iterator position)
- iterator **erase** (const_iterator first, const_iterator last)

Static Public Member Functions

- static size_t **max_size** ()

Static Public Attributes

- static size_t **_maxsize** = 100

Additional Inherited Members

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

- deque.h

4.2 Deque Class Reference

Class implemention of iterator of deque.

`#include <deque.h>`

4.2.1 Detailed Description

Class implemention of iterator of deque.

Author

couatl

Version

1.0

Date

25/11/2016

Big Boss of this project

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

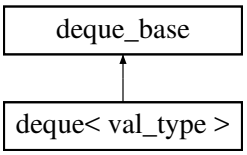
- deque.h

4.3 deque_base Class Reference

Deque.

`#include <deque.h>`

Inheritance diagram for deque_base:



Public Attributes

- size_t **_buff_size**

4.3.1 Detailed Description

Deque.

Author

couatl

Version

1.0.1

Date

25/11/2016

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

- deque.h

4.4 deque_iterator< val_type, ref, ptr > Class Template Reference

Class implemention of iterator of deque.

`#include <deque.h>`

Public Types

- typedef node< val_type > **Node**
- typedef deque_iterator< val_type, val_type &, val_type * > **iterator**
- typedef deque_iterator< val_type, const val_type &, const val_type * > **const_iterator**
- typedef deque_base **_deque**

Public Member Functions

- deque_iterator (Node *base)

  Constructs a deque from one base node.
- deque_iterator (Node *cur, Node *first, _deque *base)

  Constructor for initialazing a deque with begin but not with an end.
- deque_iterator (Node *cur=nullptr, Node *first=nullptr, Node *last=nullptr, _deque *base=nullptr)

  Default constructor.
- deque_iterator (const iterator &T)

  Copy constructor.
- val_type & operator* () const

  Dereference operator.
- deque_iterator & operator++ ()

  Operator ++.
- deque_iterator operator+ (ptrdiff_t n) const

  Operator +.
- deque_iterator operator++ (int)

  Reverse operator +.
- deque_iterator & operator+= (ptrdiff_t n)

  Operator +=.
- deque_iterator & operator-- ()

  Prefixed operator –.
- deque_iterator & operator-- (int)

  Postfixed operator –.
- deque_iterator & operator- (ptrdiff_t n) const

  Operator -.
- deque_iterator & operator-= (ptrdiff_t n)

  Operator -=.
- val_type * operator-> () const

  Operator access.
- val_type & **operator[ ]** (ptrdiff_t n) const
- deque_iterator & **operator=** (const iterator &copy)

Public Attributes

- Node * **_cur**
- Node * **_first**
- Node * **_last**
- _deque * **_base**

4.4.1 Detailed Description

template<class val_type, class ref, class ptr>
class deque_iterator< val_type, ref, ptr >

Class implemention of iterator of deque.

Author

couatl

Version

1.0

Date

25/11/2016

Implemented as Random Access Iterator

4.4.2 Constructor & Destructor Documentation

4.4.2.1 deque_iterator()

```
template<class val_type, class ref, class ptr>
deque_iterator< val_type, ref, ptr >::deque_iterator (
            const iterator & T )  [inline]
```

Copy constructor.

So basic copy constructor

Parameters

in	*T*	Object to copy

4.4.3 Member Function Documentation

4.4.3.1 operator∗()

```
template<class val_type, class ref, class ptr>
val_type& deque_iterator< val_type, ref, ptr >::operator* ( ) const  [inline]
```

Dereference operator.

Returns

Value of current node Dereferencing of _cur element of this class

4.4.3.2 operator+()

```
template<class val_type, class ref, class ptr>
deque_iterator deque_iterator< val_type, ref, ptr >::operator+ (
            ptrdiff_t n ) const  [inline]
```

Operator +.

Returns

Copy of the next iterator

Parameters

in	*n*	Number to offset your current iterator Implemented based on prefixed ++. Need to Random Access Iterator

4.4.3.3 operator++() [1/2]

```
template<class val_type, class ref, class ptr>
deque_iterator& deque_iterator< val_type, ref, ptr >::operator++ ( )  [inline]
```

Operator ++.

Returns

Object of the next iterator Creates an element if it doesn't exist for now

4.4.3.4 operator++() [2/2]

```
template<class val_type, class ref, class ptr>
deque_iterator deque_iterator< val_type, ref, ptr >::operator++ (
            int  )  [inline]
```

Reverse operator +.

Returns

Copy of the offset iterator

Parameters

in	*n*	Number to offset your current iterator Implemented based on prefixed ++. Need to Random Access Iterator

4.4.3.5 operator+=()

```
template<class val_type, class ref, class ptr>
deque_iterator& deque_iterator< val_type, ref, ptr >::operator+= (
            ptrdiff_t n )  [inline]
```

Operator +=.

Returns

Offset iterator

Parameters

in	*n*	Number to offset your current iterator Implemented based on operator +

4.4.3.6 operator-()

```
template<class val_type, class ref, class ptr>
deque_iterator& deque_iterator< val_type, ref, ptr >::operator- (
            ptrdiff_t n ) const  [inline]
```

Operator -.

Returns

Object of the offset iterator Based on prefixed operator –

4.4.3.7 operator--() [1/2]

```
template<class val_type, class ref, class ptr>
deque_iterator& deque_iterator< val_type, ref, ptr >::operator-- ( )  [inline]
```

Prefixed operator –.

Returns

Object of the previous iterator Creates an element if it doesn't exist for now, even if you're outrange (so-so)

4.4.3.8 operator--() [2/2]

```
template<class val_type, class ref, class ptr>
deque_iterator& deque_iterator< val_type, ref, ptr >::operator-- (
            int  )  [inline]
```

Postfixed operator –.

Returns

Object of the previous iterator Based on prefixed operator –

4.4.3.9 operator-=()

```
template<class val_type, class ref, class ptr>
deque_iterator& deque_iterator< val_type, ref, ptr >::operator-= (
            ptrdiff_t n )  [inline]
```

Operator -=.

Returns

Offset iterator

Parameters

in	*n*	Number to offset your current iterator Implemented based on operator -

4.4.3.10 operator->()

```
template<class val_type, class ref, class ptr>
val_type* deque_iterator< val_type, ref, ptr >::operator-> ( ) const  [inline]
```

Operator access.

Returns

Pointer to value in the current iterator Returning pointer to the value storing in the current iterator

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

- deque.h

4.5 Node Class Reference

Class of the element of deque.

```
#include <deque.h>
```

4.5.1 Detailed Description

Class of the element of deque.

Author

couatl

Version

1.0

Date

25/11/2016

So ordinar node of ordinar deque

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

- deque.h

4.6 node< val_type > Struct Template Reference

Public Types

- typedef deque_base **_deque**

Public Member Functions

- node (val_type value=0, node *next=nullptr, node *prev=nullptr, _deque *base=nullptr)
  Default constructor.
- ~node ()
  Destructor.

Public Attributes

- val_type **_value**
- node * **_next**
- node * **_prev**
- _deque * **_base**

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

- deque.h

Chapter 5

File Documentation

5.1 deque.h File Reference

Implemention of the deque, supporting iterators Created by couatl on 09.04.16.

```
#include <cstddef>
#include <iostream>
```

Classes

- class deque_base
  Deque.
- struct node< val_type >
- class deque_iterator< val_type, ref, ptr >
  Class implemention of iterator of deque.
- class deque< val_type >

Functions

- template<class val_type >
  bool **operator==** (const deque_iterator< val_type, val_type &, val_type *> &it1, const deque_iterator< val_type, val_type &, val_type *> &it2)
- template<class val_type >
  bool **operator!=** (const deque_iterator< val_type, val_type &, val_type *> &it1, const deque_iterator< val_type, val_type &, val_type *> &it2)
- template<class val_type >
  bool **operator**< (const deque_iterator< val_type, val_type &, val_type *> &it1, const deque_iterator< val_type, val_type &, val_type *> &it2)
- template<class val_type >
  bool **operator**> (const deque_iterator< val_type, val_type &, val_type *> &it1, const deque_iterator< val_type, val_type &, val_type *> &it2)
- template<class val_type >
  bool **operator<=** (const deque_iterator< val_type, val_type &, val_type *> &it1, const deque_iterator< val_type, val_type &, val_type *> &it2)

- template<class val_type >
  bool **operator>=** (const deque_iterator< val_type, val_type &, val_type *> &it1, const deque_iterator< val_type, val_type &, val_type *> &it2)
- template<class val_type >
  ptrdiff_t **operator-** (const deque_iterator< val_type, val_type &, val_type *> it1, const deque_iterator< val_type, val_type &, val_type *> it2)
- template<class val_type >
  deque_iterator< val_type, val_type &, val_type * > **operator+** (ptrdiff_t n, deque_iterator< val_type, val_type &, val_type *> it)

5.1.1 Detailed Description

Implemention of the deque, supporting iterators Created by couatl on 09.04.16.

Index