

## My Project

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	it::help::Iterable< T > Struct Template Reference . . . . .	7
4.2	it::help::Iterable< T * > Struct Template Reference . . . . .	7
4.3	it::help::Iterable< T[N]> Struct Template Reference . . . . .	7
4.4	it::help::IterTagOrder< class > Struct Template Reference . . . . .	8
4.5	it::help::IterTagOrder< std::bidirectional_iterator_tag > Struct Template Reference . . . . .	8
4.6	it::help::IterTagOrder< std::forward_iterator_tag > Struct Template Reference . . . . .	8
4.7	it::help::IterTagOrder< std::random_access_iterator_tag > Struct Template Reference . . . . .	8
4.8	it::help::SelectIterTag<... > Struct Template Reference . . . . .	8
4.9	it::help::SelectIterTag< Iter > Struct Template Reference . . . . .	9
4.10	it::help::SelectIterTag< Iter, Iters... > Struct Template Reference . . . . .	9
4.11	it::help::SelectIterTag< Iter1, Iter2 > Struct Template Reference . . . . .	9
4.12	it::UnZip< F > Struct Template Reference . . . . .	9
4.13	it::Zip< Containers > Class Template Reference . . . . .	10
4.14	it::ZipIter< Iters > Class Template Reference . . . . .	10
<b>5</b>	<b>File Documentation</b>	<b>13</b>
5.1	/home/matheus/Algoritmos/ZipIter/ZipIter/ZipIter.h File Reference . . . . .	13
	<b>Index</b>	<b>15</b>



## Chapter 1

# Hierarchical Index

### 1.1 Class Hierarchy

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

it::help::Iterable< T > . . . . .	7
it::help::Iterable< T * > . . . . .	7
it::help::Iterable< T[N]> . . . . .	7
iterator	
it::Ziplter< lters > . . . . .	10
it::help::IterTagOrder< class > . . . . .	8
it::help::IterTagOrder< std::bidirectional_iterator_tag > . . . . .	8
it::help::IterTagOrder< std::forward_iterator_tag > . . . . .	8
it::help::IterTagOrder< std::random_access_iterator_tag > . . . . .	8
it::help::SelectIterTag<... > . . . . .	8
it::help::SelectIterTag< lter > . . . . .	9
it::help::SelectIterTag< lter, SelectIterTag< lters... >::type > . . . . .	8
it::help::SelectIterTag< lter, lters... > . . . . .	9
it::help::SelectIterTag< lter1, lter2 > . . . . .	9
it::UnZip< F > . . . . .	9
it::Zip< Containers > . . . . .	10



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">it::help::Iterable&lt; T &gt;</a>	7
<a href="#">it::help::Iterable&lt; T * &gt;</a>	7
<a href="#">it::help::Iterable&lt; T[N]&gt;</a>	7
<a href="#">it::help::IterTagOrder&lt; class &gt;</a>	8
<a href="#">it::help::IterTagOrder&lt; std::bidirectional_iterator_tag &gt;</a>	8
<a href="#">it::help::IterTagOrder&lt; std::forward_iterator_tag &gt;</a>	8
<a href="#">it::help::IterTagOrder&lt; std::random_access_iterator_tag &gt;</a>	8
<a href="#">it::help::SelectIterTag&lt;... &gt;</a>	8
<a href="#">it::help::SelectIterTag&lt; Iter &gt;</a>	9
<a href="#">it::help::SelectIterTag&lt; Iter, Iters... &gt;</a>	9
<a href="#">it::help::SelectIterTag&lt; Iter1, Iter2 &gt;</a>	9
<a href="#">it::UnZip&lt; F &gt;</a>	9
<a href="#">it::Zip&lt; Containers &gt;</a>	10
<a href="#">it::ZipIter&lt; Iters &gt;</a>	10





## Chapter 3

# File Index

### 3.1 File List

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

`/home/matheus/Algoritmos/Ziplter/Ziplter/Ziplter.h`

Simple facilities to iterate through multiple containers at the same time similar to Python's `zip` . [13](#)



## Chapter 4

# Class Documentation

### 4.1 `it::help::Iterable< T >` Struct Template Reference

#### Public Types

- using **iterator** = typename T::iterator

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

- /home/matheus/Algoritmos/Ziplter/Ziplter/[Ziplter.h](#)

### 4.2 `it::help::Iterable< T * >` Struct Template Reference

#### Public Types

- using **iterator** = T \*

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

- /home/matheus/Algoritmos/Ziplter/Ziplter/[Ziplter.h](#)

### 4.3 `it::help::Iterable< T[N]>` Struct Template Reference

#### Public Types

- using **iterator** = T[N]

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

- /home/matheus/Algoritmos/Ziplter/Ziplter/[Ziplter.h](#)

#### 4.4 `it::help::IterTagOrder< class >` Struct Template Reference

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

- `/home/matheus/Algoritmos/Ziplter/Ziplter/Ziplter.h`

#### 4.5 `it::help::IterTagOrder< std::bidirectional_iterator_tag >` Struct Template Reference

##### Static Public Attributes

- static constexpr int **value** = `IterTagOrder` < `std::forward_iterator_tag` >::value + 1

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

- `/home/matheus/Algoritmos/Ziplter/Ziplter/Ziplter.h`

#### 4.6 `it::help::IterTagOrder< std::forward_iterator_tag >` Struct Template Reference

##### Static Public Attributes

- static constexpr int **value** = 0

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

- `/home/matheus/Algoritmos/Ziplter/Ziplter/Ziplter.h`

#### 4.7 `it::help::IterTagOrder< std::random_access_iterator_tag >` Struct Template Reference

##### Static Public Attributes

- static constexpr int **value** = `IterTagOrder` < `std::bidirectional_iterator_tag` >::value + 1

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

- `/home/matheus/Algoritmos/Ziplter/Ziplter/Ziplter.h`

#### 4.8 `it::help::SelectIterTag<... >` Struct Template Reference

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

- `/home/matheus/Algoritmos/Ziplter/Ziplter/Ziplter.h`

## 4.9 `it::help::SelectIterTag< Iter >` Struct Template Reference

### Public Types

- using **type** = Iter

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

- </home/matheus/Algoritmos/Ziplter/Ziplter/Ziplter.h>

## 4.10 `it::help::SelectIterTag< Iter, Iterers... >` Struct Template Reference

Inheritance diagram for `it::help::SelectIterTag< Iter, Iterers... >`:

Collaboration diagram for `it::help::SelectIterTag< Iter, Iterers... >`:

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

- </home/matheus/Algoritmos/Ziplter/Ziplter/Ziplter.h>

## 4.11 `it::help::SelectIterTag< Iter1, Iter2 >` Struct Template Reference

### Public Types

- using **type** = std::conditional\_t< ((IterTagOrder\_v< Iter1 >) < (IterTagOrder\_v< Iter2 >)), Iter1, Iter2 >

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

- </home/matheus/Algoritmos/Ziplter/Ziplter/Ziplter.h>

## 4.12 `it::UnZip< F >` Struct Template Reference

### Public Member Functions

- **UnZip** (const F &f=F())
- template<typename... Args, typename Indexes = std::make\_index\_sequence<sizeof...(Args)>>  
void **operator()** (std::tuple< Args... > &&tup)
- template<typename... Args, std::size\_t... Is>  
void **unpack** (std::tuple< Args... > &tup, std::index\_sequence< Is... >)

## Public Attributes

- **F f**

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

- </home/matheus/Algoritmos/Ziplter/Ziplter/Ziplter.h>

## 4.13 `it::Zip< Containers >` Class Template Reference

### Public Types

- using **value\_type** = `std::tuple< Containers... >`
- using **iterator** = `Ziplter< typename help::Iterable< std::decay_t< Containers > >::iterator... >`
- using **const\_iterator** = `iterator`

### Public Member Functions

- **Zip** (`Containers...conts`)
- **iterator begin** ()
- **const\_iterator begin** () const
- **iterator end** ()
- **const\_iterator end** () const
- `template<class Tag = typename iterator::iterator_category, std::enable_if_t< std::is_same< Tag, std::random_access_iterator_tag >↔::value, int > = 0>`  
`decltype(auto) operator[] (std::size_t pos)`
- `template<class Tag = typename iterator::iterator_category, std::enable_if_t< std::is_same< Tag, std::random_access_iterator_tag >↔::value, int > = 0>`  
`decltype(auto) operator[] (std::size_t pos) const`
- `std::size_t size () const`

### Static Public Attributes

- static constexpr `std::size_t containersSize = sizeof... (Containers)`

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

- </home/matheus/Algoritmos/Ziplter/Ziplter/Ziplter.h>

## 4.14 `it::Ziplter< Iters >` Class Template Reference

Inheritance diagram for `it::Ziplter< Iters >`:

Collaboration diagram for `it::Ziplter< Iters >`:

## Public Types

- using **Base** = std::iterator< help::SelectIterTag\_t< typename std::iterator\_traits< Iters >::iterator\_category... >, std::tuple< Iters... > >
- using **value\_type** = typename Base::value\_type
- using **reference** = typename Base::reference
- using **difference\_type** = typename Base::difference\_type
- using **iterator\_category** = typename Base::iterator\_category

## Public Member Functions

- **Ziplter** (Iters...iterators)
- **Ziplter** (std::tuple< Iters... > iters)
- **Ziplter** & **operator++** ()
- **Ziplter** **operator++** (int)
- template<class Tag = iterator\_category, std::enable\_if\_t< std::is\_same< Tag, std::bidirectional\_iterator\_tag >::value, int > = 0>  
**Ziplter** & **operator--** ()
- template<class Tag = iterator\_category, std::enable\_if\_t< std::is\_same< Tag, std::bidirectional\_iterator\_tag >::value, int > = 0>  
**Ziplter** **operator--** (int)
- template<class Tag = iterator\_category, std::enable\_if\_t< std::is\_same< Tag, std::random\_access\_iterator\_tag >::value, int > = 0>  
**Ziplter** **operator+** (int inc) const
- template<class Tag = iterator\_category, std::enable\_if\_t< std::is\_same< Tag, std::random\_access\_iterator\_tag >::value, int > = 0>  
**Ziplter** **operator-** (int inc) const
- difference\_type **operator+** (const **Ziplter** &zp) const
- difference\_type **operator-** (const **Ziplter** &zp) const
- template<typename Iter >  
bool **operator==** (const Iter &iter) const
- template<typename Iter >  
bool **operator!=** (const Iter &iter) const
- decltype(auto) **operator\*** ()
- decltype(auto) **operator\*** () const

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

- /home/matheus/Algoritmos/Ziplter/Ziplter/[Ziplter.h](#)





## Chapter 5

# File Documentation

### 5.1 /home/matheus/Algoritmos/Ziplter/Ziplter/Ziplter.h File Reference

Simple facilities to iterate through multiple containers at the same time similar to Python's zip.

```
#include <type_traits>
#include <tuple>
#include <iterator>
Include dependency graph for Ziplter.h:
```



# Index

/home/matheus/Algoritmos/Ziplter/Ziplter/Ziplter.h, [13](#)

it::UnZip< F >, [9](#)

it::Zip< Containers >, [10](#)

it::Ziplter< lters >, [10](#)

it::help::IterTagOrder< class >, [8](#)

it::help::IterTagOrder< std::bidirectional\_iterator\_tag >, [8](#)

it::help::IterTagOrder< std::forward\_iterator\_tag >, [8](#)

it::help::IterTagOrder< std::random\_access\_iterator\_tag >, [8](#)

it::help::Iterable< T >, [7](#)

it::help::Iterable< T \* >, [7](#)

it::help::Iterable< T[N]>, [7](#)

it::help::SelectIterTag< lter >, [9](#)

it::help::SelectIterTag< lter, lters... >, [9](#)

it::help::SelectIterTag< lter1, lter2 >, [9](#)

it::help::SelectIterTag<... >, [8](#)