

## My Project

Generated by Doxygen 1.9.4



<b>1 Class Index</b>	<b>1</b>
1.1 Class List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 cmdargs Struct Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Data Documentation	5
3.1.2.1 verbose	5
3.1.2.2 x	5
3.1.2.3 y	5
<b>4 File Documentation</b>	<b>7</b>
4.1 addpp.c File Reference	7
4.1.1 Detailed Description	8
4.1.2 Macro Definition Documentation	8
4.1.2.1 VERBOSE_LEN	8
4.1.3 Function Documentation	8
4.1.3.1 main()	8
4.2 cmdargs.c File Reference	8
4.2.1 Function Documentation	9
4.2.1.1 cmdargs_parse()	9
4.3 cmdargs.h File Reference	10
4.3.1 Function Documentation	10
4.3.1.1 cmdargs_parse()	10
4.4 cmdargs.h	11
4.5 cmdargs_test.c File Reference	11
4.5.1 Detailed Description	11
4.5.2 Function Documentation	11
4.5.2.1 main()	12
4.6 core.c File Reference	12
4.6.1 Detailed Description	12
4.6.2 Function Documentation	12
4.6.2.1 add()	12
4.7 core.h File Reference	13
4.7.1 Function Documentation	14
4.7.1.1 add()	14
4.8 core.h	14
4.9 core_test.c File Reference	14
4.9.1 Detailed Description	15
4.9.2 Function Documentation	15

4.9.2.1 main() . . . . .	15
--------------------------	----

<b>Index</b>	<b>17</b>
--------------	-----------

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">cmdargs</a> . . . . .	5
-----------------------------------	---



## Chapter 2

# File Index

### 2.1 File List

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

<a href="#">addpp.c</a>	7
<a href="#">cmdargs.c</a>	8
<a href="#">cmdargs.h</a>	10
<a href="#">cmdargs_test.c</a>	11
<a href="#">core.c</a>	12
<a href="#">core.h</a>	13
<a href="#">core_test.c</a>	14





## Chapter 3

# Class Documentation

### 3.1 cmdargs Struct Reference

```
#include <cmdargs.h>
```

#### Public Attributes

- int [x](#)
- int [y](#)
- int [verbose](#)

#### 3.1.1 Detailed Description

Holds all the possible command line arguments

#### 3.1.2 Member Data Documentation

##### 3.1.2.1 verbose

```
int cmdargs::verbose
```

An flag that if specified requests verbose output

##### 3.1.2.2 x

```
int cmdargs::x
```

The first summand

##### 3.1.2.3 y

```
int cmdargs::y
```

The second summand

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

- [cmdargs.h](#)

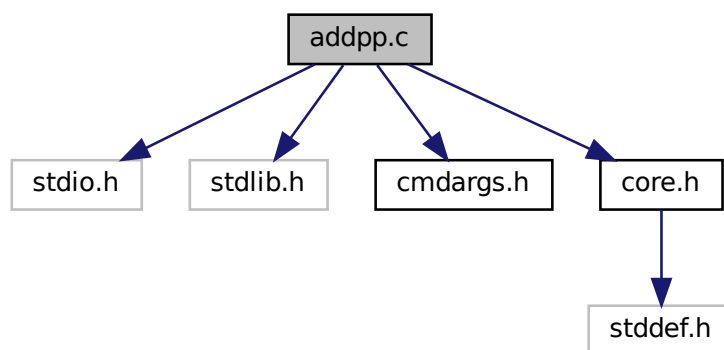


## Chapter 4

# File Documentation

### 4.1 addpp.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "cmdargs.h"
#include "core.h"
Include dependency graph for addpp.c:
```



#### Macros

- `#define` `VERBOSE_LEN` 100

#### Functions

- `int` `main` (`int argc`, `char **argv`)

### 4.1.1 Detailed Description

This is the main driver of the program, i.e., the program, which is then used by the user.

### 4.1.2 Macro Definition Documentation

#### 4.1.2.1 VERBOSE\_LEN

```
#define VERBOSE_LEN 100
```

The length of the verbose buffer.

Often, it is better not to use define like here but const variables. The define is used for demonstration purposes.

### 4.1.3 Function Documentation

#### 4.1.3.1 main()

```
int main (
    int argc,
    char ** argv )
```

The program's main entry point.

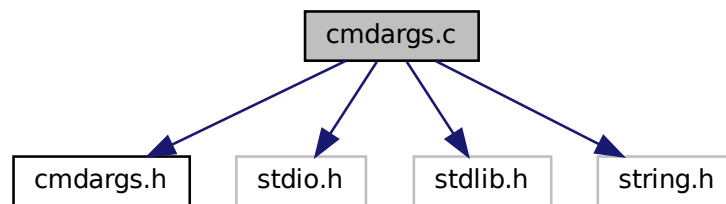
##### Parameters

<i>argc</i>	number of arguments given by the user
<i>argv</i>	the argument vector

## 4.2 cmdargs.c File Reference

```
#include "cmdargs.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for cmdargs.c:



## Functions

- int `cmdargs_parse` (struct `cmdargs` \*`cmdargs`, int `argc`, char \*`argv`[])

### 4.2.1 Function Documentation

#### 4.2.1.1 cmdargs\_parse()

```
int cmdargs_parse (  
    struct cmdargs * cmdargs,  
    int argc,  
    char * argv[] )
```

Parse the given command line arguments into the `cmdargs` struct.

#### Parameters

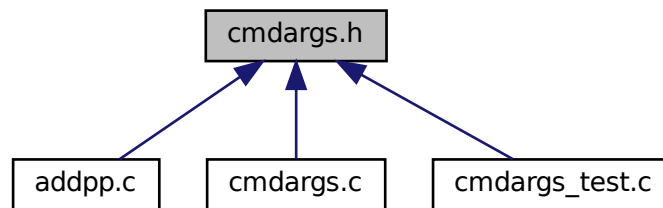
<i>cmdargs</i>	
<i>argc</i>	the size of <code>argv</code> as supplied in <code>main()</code> .
<i>argv</i>	the the array of arguments as supplied in <code>main()</code> .

#### Returns

1 on success, 0 on failure.

## 4.3 cmdargs.h File Reference

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



### Classes

- struct [cmdargs](#)

### Functions

- int [cmdargs\\_parse](#) (struct [cmdargs](#) \*[cmdargs](#), int *argc*, char \**argv*[ ])

#### 4.3.1 Function Documentation

##### 4.3.1.1 cmdargs\_parse()

```
int cmdargs_parse (  
    struct cmdargs * cmdargs,  
    int argc,  
    char * argv[ ] )
```

Parse the given command line arguments into the `cmdargs` struct.

#### Parameters

<i>cmdargs</i>	
<i>argc</i>	the size of <i>argv</i> as supplied in <a href="#">main()</a> .
<i>argv</i>	the the array of arguments as supplied in <a href="#">main()</a> .

### Returns

1 on success, 0 on failure.

## 4.4 cmdargs.h

[Go to the documentation of this file.](#)

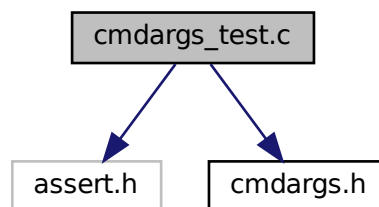
```
1
5 #ifndef _CMDARGS_H_
6 #define _CMDARGS_H_
7
9 struct cmdargs
10 {
11     int x;
12     int y;
13     int verbose;
14 };
15
24 int cmdargs_parse(struct cmdargs *cmdargs, int argc, char *argv[]);
25
26 #endif
```

## 4.5 cmdargs\_test.c File Reference

```
#include <assert.h>
```

```
#include "cmdargs.h"
```

Include dependency graph for cmdargs\_test.c:



### Functions

- int [main](#) (int argc, char \*\*argv)

#### 4.5.1 Detailed Description

Test for functions in [core.c](#)

#### 4.5.2 Function Documentation

#### 4.5.2.1 main()

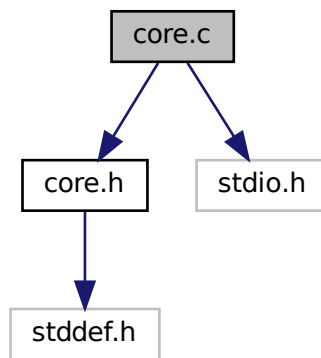
```
int main (
    int argc,
    char ** argv )
```

Main entry for the test.

## 4.6 core.c File Reference

```
#include "core.h"
#include <stdio.h>
```

Include dependency graph for core.c:



### Functions

- int [add](#) (int x, int y, char \*verbose, size\_t verbose\_len)

#### 4.6.1 Detailed Description

Implements the core of the algorithm.

#### 4.6.2 Function Documentation

##### 4.6.2.1 add()

```
int add (
    int x,
    int y,
    char * verbose,
    size_t verbose_len )
```

This is a simple function to add two integer values. It main purpose is to demonstrate the testing.



## Parameters

<i>x</i>	first summand
<i>y</i>	second summand
<i>verbose</i>	defines the pointer to a buffer where verbose output is written to
<i>verbose_len</i>	defines the length of the verbose buffer

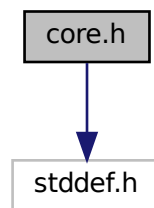
## Returns

the sum of x and y

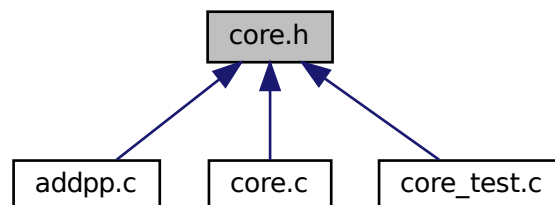
## 4.7 core.h File Reference

```
#include <stddef.h>
```

Include dependency graph for core.h:



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



## Functions

- int [add](#)(int x, int y, char \*verbose, size\_t verbose\_len)

## 4.7.1 Function Documentation

### 4.7.1.1 add()

```
int add (
    int x,
    int y,
    char * verbose,
    size_t verbose_len )
```

This is a simple function to add two integer values. It main purpose is to demonstrate the testing.

#### Parameters

<i>x</i>	first summand
<i>y</i>	second summand
<i>verbose</i>	defines the pointer to a buffer where verbose output is written to
<i>verbose_len</i>	defines the length of the verbose buffer

#### Returns

the sum of x and y

## 4.8 core.h

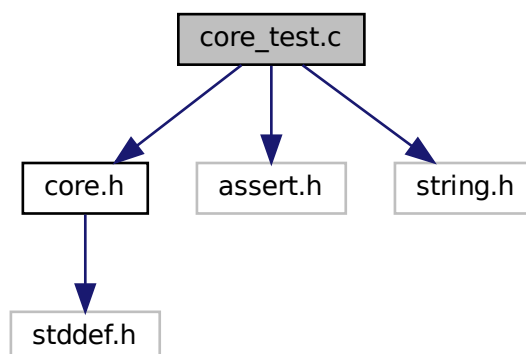
[Go to the documentation of this file.](#)

```
1
4 #ifndef _CORE_H
5 #define _CORE_H
6
7 #include <stddef.h>
8
21 int add(int x, int y, char *verbose, size_t verbose_len);
22
23 #endif
```

## 4.9 core\_test.c File Reference

```
#include "core.h"
#include <assert.h>
#include <string.h>
```

Include dependency graph for core\_test.c:



## Functions

- int [main](#) (int argc, char \*\*argv)

### 4.9.1 Detailed Description

Test for functions in [core.c](#)

### 4.9.2 Function Documentation

#### 4.9.2.1 main()

```
int main (  
    int argc,  
    char ** argv )
```

Main entry for the test.



# Index

- add
  - core.c, [12](#)
  - core.h, [14](#)
- addpp.c, [7](#)
  - main, [8](#)
  - VERBOSE\_LEN, [8](#)
- cmdargs, [5](#)
  - verbose, [5](#)
  - x, [5](#)
  - y, [5](#)
- cmdargs.c, [8](#)
  - cmdargs\_parse, [9](#)
- cmdargs.h, [10](#)
  - cmdargs\_parse, [10](#)
- cmdargs\_parse
  - cmdargs.c, [9](#)
  - cmdargs.h, [10](#)
- cmdargs\_test.c, [11](#)
  - main, [11](#)
- core.c, [12](#)
  - add, [12](#)
- core.h, [13](#)
  - add, [14](#)
- core\_test.c, [14](#)
  - main, [15](#)
- main
  - addpp.c, [8](#)
  - cmdargs\_test.c, [11](#)
  - core\_test.c, [15](#)
- verbose
  - cmdargs, [5](#)
- VERBOSE\_LEN
  - addpp.c, [8](#)
- x
  - cmdargs, [5](#)
- y
  - cmdargs, [5](#)