

commandinterpreter

Generated by Doxygen 1.8.14

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	cmd Union Reference	5
3.2	cmd_table_entry Struct Reference	5
3.2.1	Detailed Description	5
3.2.2	Member Data Documentation	6
3.2.2.1	next	6
3.2.2.2	value	6
4	File Documentation	7
4.1	cmdinterpreter.h File Reference	7
4.1.1	Detailed Description	8
4.1.2	Variable Documentation	8
4.1.2.1	args	8
4.1.2.2	doc	8
4.1.2.3	entries	8
4.1.2.4	func	8
	Index	9

Chapter 1

Class Index

1.1 Class List

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

cmd	5
cmd_table_entry Entry in the hashtable used to store the commands	5

Chapter 2

File Index

2.1 File List

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

cmdinterpreter.h	Interpreter for custom command. Might be used in CL-applications	7
----------------------------------	--	---

Chapter 3

Class Documentation

3.1 cmd Union Reference

Public Attributes

- int **i**
- float **f**
- char **c**
- char * **s**
- void * **v**

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

- [cmdinterpreter.h](#)

3.2 cmd_table_entry Struct Reference

an entry in the hashtable used to store the commands

```
#include <cmdinterpreter.h>
```

Public Attributes

- char * **key**
- [cmd](#) * **value**
- struct [cmd_table_entry](#) * **next**

3.2.1 Detailed Description

an entry in the hashtable used to store the commands

3.2.2 Member Data Documentation

3.2.2.1 next

```
struct cmd\_table\_entry* cmd_table_entry::next
```

< actual value in form of a pointer to the command

3.2.2.2 value

```
cmd* cmd_table_entry::value
```

< key of the entrie/command

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

- [cmdinterpreter.h](#)

Chapter 4

File Documentation

4.1 cmdinterpreter.h File Reference

interpreter for custom command. Might be used in CL-applications.

Classes

- union `cmd`
- struct `cmd_table_entry`
an entry in the hashtable used to store the commands

Functions

- void `cmd_table_create` (void)
- void `command_add` (void)
- void `command_remove` (void)
- void `command_get_args` (void)
- void `command_execute` (void)
- void `command_interprete` (void)

Variables

- ```
struct {
 char * name
 int(* func)(cmd_arg *args)
 char * args
 char * doc
} cmd
```

  
*structure containing the details of a command*
- struct `cmd_table_entry` **cmd\_table\_entry**
- ```
struct {  
    int size  
    cmd_table_entry ** entries  
} cmd_table
```

The struct for storing the commands. It works as a hashtable.

4.1.1 Detailed Description

interpreter for custom command. Might be used in CL-applications.

Author

Torsten Lehmann

Date

2018-05-19

4.1.2 Variable Documentation

4.1.2.1 args

`char* args`

< function-pointer to the function the command should call

4.1.2.2 doc

`char* doc`

< list of arguments the command requires

4.1.2.3 entries

`cmd_table_entry** entries`

< size of the hashtable

4.1.2.4 func

`int (* func) (cmd_arg *args)`

< name of the command

Index

- args
 - [cmdinterpreter.h](#), [8](#)
- cmd, [5](#)
- cmd_table_entry, [5](#)
 - [next](#), [6](#)
 - [value](#), [6](#)
- cmdinterpreter.h, [7](#)
 - [args](#), [8](#)
 - [doc](#), [8](#)
 - [entries](#), [8](#)
 - [func](#), [8](#)
- doc
 - [cmdinterpreter.h](#), [8](#)
- entries
 - [cmdinterpreter.h](#), [8](#)
- func
 - [cmdinterpreter.h](#), [8](#)
- next
 - [cmd_table_entry](#), [6](#)
- value
 - [cmd_table_entry](#), [6](#)