gds

Generated by Doxygen 1.8.1.2

Sat Nov 8 2014 15:28:03

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

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

Chapter 3

Data Structure Documentation

3.1 gdt_generic_datatype Struct Reference

Data Fields

- enum gds_datatype **type**
- gds_cfunc **compfunc**
- union {
    char **c**
    unsigned char **uc**
    signed char **sc**
    int **i**
    unsigned int **ui**
    long **l**
    unsigned long **ul**
    long long int **ll**
    unsigned long long int **ull**
    size_t **st**
    double **d**
    char ∗ **pc**
    void ∗ **p**
  } **data**

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

- include/private/gdt.h

3.2 hms Struct Reference

Data Fields

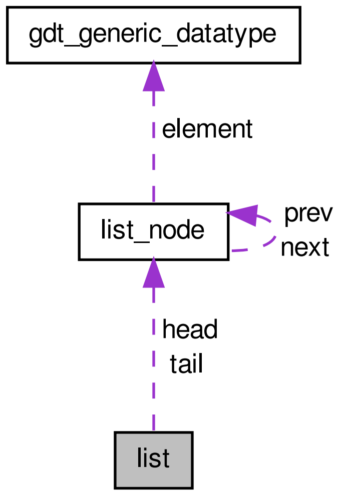
- int **hour**
- int **minute**
- int **second**

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

- tests/test_list.c
- tests/test_vector.c

3.3 list Struct Reference

Collaboration diagram for list:



Data Fields

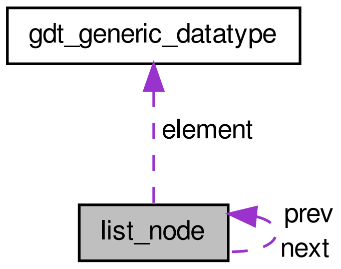
- size_t **length**
- enum gds_datatype **type**
- gds_cfunc **compfunc**
- struct list_node ∗ **head**
- struct list_node ∗ **tail**
- bool **free_on_destroy**
- bool **exit_on_error**

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

- src/list.c

3.4 list_node Struct Reference

Collaboration diagram for list_node:



Data Fields

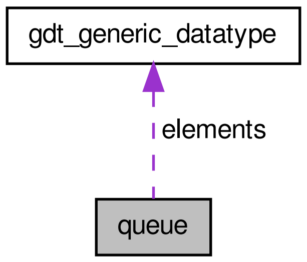
- struct gdt_generic_datatype **element**
- struct list_node ∗ **prev**
- struct list_node ∗ **next**

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

- src/list.c

3.5 queue Struct Reference

Collaboration diagram for queue:



Data Fields

- size_t **front**

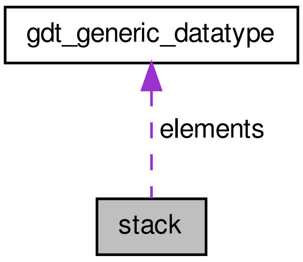
- size_t **back**
- size_t **capacity**
- size_t **size**
- enum gds_datatype **type**
- struct gdt_generic_datatype ∗ **elements**
- bool **resizable**
- bool **free_on_destroy**
- bool **exit_on_error**

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

- src/queue.c

3.6 stack Struct Reference

Collaboration diagram for stack:



Data Fields

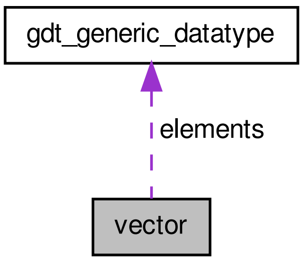
- size_t **top**
- size_t **capacity**
- enum gds_datatype **type**
- struct gdt_generic_datatype ∗ **elements**
- bool **resizable**
- bool **free_on_destroy**
- bool **exit_on_error**

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

- src/stack.c

3.7 vector Struct Reference

Collaboration diagram for vector:



Data Fields

- size_t **length**
- size_t **capacity**
- enum gds_datatype **type**
- struct gdt_generic_datatype $*$ **elements**
- int($*$ **compfunc**)(const void $*$, const void $*$)
- bool **free_on_destroy**
- bool **exit_on_error**

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

- src/vector.c

Chapter 4

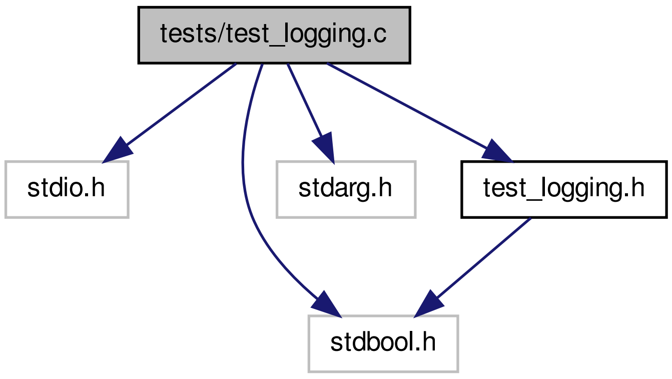
File Documentation

4.1 tests/test_logging.c File Reference

Implementation of unit test logging functionality.

```
#include <stdio.h>
#include <stdbool.h>
#include <stdarg.h>
#include "test_logging.h"
```

Include dependency graph for test_logging.c:



Functions

- void tests_log_test (const bool success, const char ∗fmt,...)

  Logs the result of a unit test.
- int tests_get_total_tests (void)

  Returns the total number of tests run.
- int tests_get_successes (void)

  Returns the total number of successful tests.
- int tests_get_failures (void)

  Returns the total number of failed tests.

4.1.1 Detailed Description

Implementation of unit test logging functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. http://www.gnu.org/licenses/

4.1.2 Function Documentation

4.1.2.1 int tests_get_failures (void)

Returns the total number of failed tests.

Returns

The total number of failed tests.

4.1.2.2 int tests_get_successes (void)

Returns the total number of successful tests.

Returns

The total number of successful tests.

4.1.2.3 int tests_get_total_tests (void)

Returns the total number of tests run.

Returns

The total number of tests run.

4.1.2.4 void tests_log_test (const bool *success*, const char $*$ *fmt*, ...)

Logs the result of a unit test.

Parameters

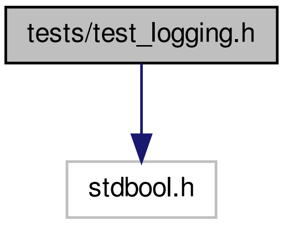
success	`true` if the test succeeded, `false` otherwise.
fmt	Format string for failure message.
...	Arguements to format string.

4.2 tests/test_logging.h File Reference

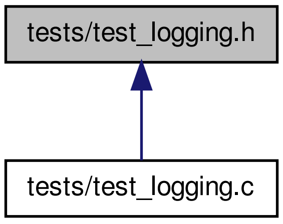
Interface to unit test logging functionality.

```
#include <stdbool.h>
```

Include dependency graph for test_logging.h:



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



Functions

- void tests_log_test (const bool success, const char ∗fmt,...)

  Logs the result of a unit test.
- int tests_get_total_tests (void)

  Returns the total number of tests run.
- int tests_get_successes (void)

  Returns the total number of successful tests.
- int tests_get_failures (void)

  Returns the total number of failed tests.

4.2.1 Detailed Description

Interface to unit test logging functionality.

Author

Paul Griffiths

Copyright

4.2.2 Function Documentation

4.2.2.1 int tests_get_failures (void)

Returns the total number of failed tests.

Returns

The total number of failed tests.

4.2.2.2 int tests_get_successes (void)

Returns the total number of successful tests.

Returns

The total number of successful tests.

4.2.2.3 int tests_get_total_tests (void)

Returns the total number of tests run.

Returns

The total number of tests run.

4.2.2.4 void tests_log_test (const bool *success*, const char ∗ *fmt*, ...)

Logs the result of a unit test.

Parameters

success	`true` if the test succeeded, `false` otherwise.
fmt	Format string for failure message.
...	Arguements to format string.

Index