

## GCCTrace

Generated by Doxygen 1.8.3.1

Thu Apr 17 2014 22:35:47



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	call_stack Struct Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Member Data Documentation . . . . .	5
3.1.2.1	frames . . . . .	5
3.1.2.2	num_frames . . . . .	5
3.2	stack_frame Struct Reference . . . . .	5
3.2.1	Detailed Description . . . . .	6
3.2.2	Member Data Documentation . . . . .	6
3.2.2.1	call_site . . . . .	6
3.2.2.2	this_fn . . . . .	6
3.2.2.3	thread . . . . .	6
3.2.2.4	time . . . . .	6
3.2.2.5	used_bytes . . . . .	6
<b>4</b>	<b>File Documentation</b>	<b>7</b>
4.1	include/gcctracer.h File Reference . . . . .	7
4.1.1	Detailed Description . . . . .	7
4.1.2	Typedef Documentation . . . . .	8
4.1.2.1	call_stack . . . . .	8
4.1.2.2	stack_frame . . . . .	8
4.1.3	Function Documentation . . . . .	8
4.1.3.1	_gcc_trace_clone_current_call_stack . . . . .	8
4.1.3.2	_gcc_trace_dump_history_buffer . . . . .	8
4.1.3.3	_gcc_trace_free_call_stack . . . . .	8
4.1.3.4	_gcc_trace_get_heap_memory . . . . .	9

4.1.3.5 <code>_gcc_trace_print_call_stack</code> . . . . .	9
--	---

<b>Index</b>	<b>9</b>
--------------	----------

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">call_stack</a>	Full calling stack . . . . .	5
<a href="#">stack_frame</a>	Stack frame structure . . . . .	5



## Chapter 2

# File Index

### 2.1 File List

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

include/ <a href="#">gcctracer.h</a>	
Main file to include gcctrace's functionalities . . . . .	<a href="#">7</a>





## Chapter 3

# Class Documentation

### 3.1 `call_stack` Struct Reference

Full calling stack.

```
#include <gcctracer.h>
```

#### Public Attributes

- unsigned int [num\\_frames](#)
- [stack\\_frame](#) \* [frames](#)

#### 3.1.1 Detailed Description

Full calling stack.

Structure that contains a full calling stack

#### 3.1.2 Member Data Documentation

##### 3.1.2.1 `stack_frame*` `call_stack::frames`

Stack frames array

##### 3.1.2.2 `unsigned int` `call_stack::num_frames`

How deep is the stack

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

- [include/gcctracer.h](#)

### 3.2 `stack_frame` Struct Reference

Stack frame structure.

```
#include <gcctracer.h>
```

## Public Attributes

- unsigned long int [time](#)
- unsigned long int [thread](#)
- unsigned long int [used\\_bytes](#)
- void \* [this\\_fn](#)
- void \* [call\\_site](#)

### 3.2.1 Detailed Description

Stack frame structure.

Stack frame containing the traced data

### 3.2.2 Member Data Documentation

#### 3.2.2.1 void\* `stack_frame::call_site`

Place in the code where `this_fn` was called

#### 3.2.2.2 void\* `stack_frame::this_fn`

Pointer to the invoked function

#### 3.2.2.3 unsigned long int `stack_frame::thread`

Thread ID

#### 3.2.2.4 unsigned long int `stack_frame::time`

Timestamp

#### 3.2.2.5 unsigned long int `stack_frame::used_bytes`

Allocated memory in bytes

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

- [include/gcctracer.h](#)

# Chapter 4

## File Documentation

### 4.1 include/gcctracer.h File Reference

Main file to include gcctrace's functionalities.

```
#include <stdlib.h>
```

#### Classes

- struct [stack\\_frame](#)  
*Stack frame structure.*
- struct [call\\_stack](#)  
*Full calling stack.*

#### Typedefs

- typedef struct [stack\\_frame](#) [stack\\_frame](#)  
*Stack frame structure.*
- typedef struct [call\\_stack](#) [call\\_stack](#)  
*Full calling stack.*

#### Functions

- void [\\_gcc\\_trace\\_clone\\_current\\_call\\_stack](#) ([call\\_stack](#) \*stack)
- void [\\_gcc\\_trace\\_free\\_call\\_stack](#) ([call\\_stack](#) \*stack)
- void [\\_gcc\\_trace\\_print\\_call\\_stack](#) ([call\\_stack](#) \*stack)
- void [\\_gcc\\_trace\\_dump\\_history\\_buffer](#) (const char \*file\_name)
- unsigned long int [\\_gcc\\_trace\\_get\\_heap\\_memory](#) ()

#### 4.1.1 Detailed Description

Main file to include gcctrace's functionalities.

#### Author

Renato Grottesi

Date

7 Apr 2014

## 4.1.2 Typedef Documentation

### 4.1.2.1 typedef struct call\_stack call\_stack

Full calling stack.

Structure that contains a full calling stack

### 4.1.2.2 typedef struct stack\_frame stack\_frame

Stack frame structure.

Stack frame containing the traced data

## 4.1.3 Function Documentation

### 4.1.3.1 void \_gcc\_trace\_clone\_current\_call\_stack ( call\_stack \* stack )

Copy the current call stack inside the input parameter. This function allocates stack's internal memory. Please call `_gcc_trace_free_call_stack` to free the internal memory.

Parameters

<i>stack</i>	The call stack object where to clone the current call stack
--------------	---

See Also

[\\_gcc\\_trace\\_free\\_call\\_stack](#)

### 4.1.3.2 void \_gcc\_trace\_dump\_history\_buffer ( const char \* file\_name )

Dump the internal circular buffer containing the history of the last n function invocations, where n is an hardcoded size for the buffer capacity.

Parameters

<i>file_name</i>	The name of the file where to dump the data
------------------	---

### 4.1.3.3 void \_gcc\_trace\_free\_call\_stack ( call\_stack \* stack )

Free the internal memory allocated by `_gcc_trace_clone_current_call_stack`

Parameters

<i>stack</i>	The call stack object to free
--------------	-------------------------------

See Also

[\\_gcc\\_trace\\_clone\\_current\\_call\\_stack](#)

#### 4.1.3.4 unsigned long int `_gcc_trace_get_heap_memory ( )`

Return the total heap memory allocated

##### Returns

total bytes allocated in heap

#### 4.1.3.5 void `_gcc_trace_print_call_stack ( call_stack * stack )`

Prints a call stack cloned by `_gcc_trace_clone_current_call_stack`.

##### Parameters

<i>stack</i>	The stack to print in stderr
--------------	------------------------------

##### See Also

[\\_gcc\\_trace\\_clone\\_current\\_call\\_stack](#)

# Index

- [\\_gcc\\_trace\\_clone\\_current\\_call\\_stack](#)  
gcctracer.h, [8](#)
  - [\\_gcc\\_trace\\_dump\\_history\\_buffer](#)  
gcctracer.h, [8](#)
  - [\\_gcc\\_trace\\_free\\_call\\_stack](#)  
gcctracer.h, [8](#)
  - [\\_gcc\\_trace\\_get\\_heap\\_memory](#)  
gcctracer.h, [8](#)
  - [\\_gcc\\_trace\\_print\\_call\\_stack](#)  
gcctracer.h, [9](#)
- call\_site
  - [stack\\_frame](#), [6](#)
- call\_stack, [5](#)
  - [frames](#), [5](#)
  - gcctracer.h, [8](#)
  - [num\\_frames](#), [5](#)
- frames
  - [call\\_stack](#), [5](#)
- gcctracer.h
  - [\\_gcc\\_trace\\_clone\\_current\\_call\\_stack](#), [8](#)
  - [\\_gcc\\_trace\\_dump\\_history\\_buffer](#), [8](#)
  - [\\_gcc\\_trace\\_free\\_call\\_stack](#), [8](#)
  - [\\_gcc\\_trace\\_get\\_heap\\_memory](#), [8](#)
  - [\\_gcc\\_trace\\_print\\_call\\_stack](#), [9](#)
  - [call\\_stack](#), [8](#)
  - [stack\\_frame](#), [8](#)
- include/gcctracer.h, [7](#)
- num\_frames
  - [call\\_stack](#), [5](#)
- stack\_frame, [5](#)
  - [call\\_site](#), [6](#)
  - gcctracer.h, [8](#)
  - [this\\_fn](#), [6](#)
  - [thread](#), [6](#)
  - [time](#), [6](#)
  - [used\\_bytes](#), [6](#)
- this\_fn
  - [stack\\_frame](#), [6](#)
- thread
  - [stack\\_frame](#), [6](#)
- time
  - [stack\\_frame](#), [6](#)
- used\_bytes
  - [stack\\_frame](#), [6](#)