

# Phone Firewall Reference Manual

v0.01

Generated by Doxygen 1.5.4

Thu Nov 6 10:42:10 2008



# Contents

<b>1</b>	<b>Phone Firewall Data Structure Index</b>	<b>1</b>
1.1	Phone Firewall Data Structures . . . . .	1
<b>2</b>	<b>Phone Firewall File Index</b>	<b>3</b>
2.1	Phone Firewall File List . . . . .	3
<b>3</b>	<b>Phone Firewall Data Structure Documentation</b>	<b>5</b>
3.1	Entry Struct Reference . . . . .	5
3.2	entry Struct Reference . . . . .	7
<b>4</b>	<b>Phone Firewall File Documentation</b>	<b>9</b>
4.1	libphonefirewall.h File Reference . . . . .	9
4.2	logfile.c File Reference . . . . .	13
4.3	logfile.h File Reference . . . . .	15
4.4	pf_administration.c File Reference . . . . .	18
4.5	pf_administration.h File Reference . . . . .	23
4.6	pf_search.c File Reference . . . . .	27
4.7	pf_search.h File Reference . . . . .	31



# Chapter 1

## Phone Firewall Data Structure Index

### 1.1 Phone Firewall Data Structures

Here are the data structures with brief descriptions:

<a href="#">Entry</a> . . . . .	<a href="#">5</a>
<a href="#">entry</a> (Includes all informations for an <a href="#">entry</a> ) . . . . .	<a href="#">7</a>



# Chapter 2

## Phone Firewall File Index

### 2.1 Phone Firewall File List

Here is a list of all files with brief descriptions:

<a href="#">libphonefirewall.h</a> (API of the phone firewall ) . . . . .	9
<a href="#">logfile.c</a> . . . . .	13
<a href="#">logfile.h</a> . . . . .	15
<a href="#">pf_administration.c</a> . . . . .	18
<a href="#">pf_administration.h</a> . . . . .	23
<a href="#">pf_search.c</a> . . . . .	27
<a href="#">pf_search.h</a> . . . . .	31





## Chapter 3

# Phone Firewall Data Structure Documentation

### 3.1 Entry Struct Reference

```
#include <libphonefirewall.h>
```

Collaboration diagram for Entry:



#### Data Fields

- int [country\\_code](#)
- int [area\\_code](#)
- unsigned long long [number](#)
- char \* [name](#)
- char \* [reason](#)
- int [priority](#)
- struct [Entry](#) \* [next](#)

#### 3.1.1 Detailed Description

Definition at line 60 of file libphonefirewall.h.

#### 3.1.2 Field Documentation

##### 3.1.2.1 int Entry::country\_code

Definition at line 61 of file libphonefirewall.h.

Referenced by [check\\_entry\(\)](#), [evaluate\\_stmt\(\)](#), [find\\_entry\(\)](#), and [insert\\_into\\_list\(\)](#).

### 3.1.2.2 `int Entry::area_code`

Definition at line 62 of file libphonefirewall.h.

Referenced by `check_entry()`, `evaluate_stmt()`, `find_entry()`, and `insert_into_list()`.

### 3.1.2.3 `unsigned long long Entry::number`

Definition at line 63 of file libphonefirewall.h.

Referenced by `check_entry()`, `evaluate_stmt()`, `find_entry()`, and `insert_into_list()`.

### 3.1.2.4 `char* Entry::name`

Definition at line 64 of file libphonefirewall.h.

Referenced by `find_entry()`, and `insert_into_list()`.

### 3.1.2.5 `char* Entry::reason`

Definition at line 65 of file libphonefirewall.h.

Referenced by `find_entry()`, and `insert_into_list()`.

### 3.1.2.6 `int Entry::priority`

Definition at line 66 of file libphonefirewall.h.

Referenced by `check_entry()`, `evaluate_stmt()`, and `find_entry()`.

### 3.1.2.7 `struct Entry* Entry::next` [read]

Definition at line 67 of file libphonefirewall.h.

Referenced by `insert_into_list()`.

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

- [libphonefirewall.h](#)

## 3.2 entry Struct Reference

Includes all informations for an [entry](#).

```
#include <libphonefirewall.h>
```

### 3.2.1 Detailed Description

Includes all informations for an [entry](#).

The struct which includes all information about entries (black- and whitelist).

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

- [libphonefirewall.h](#)



## Chapter 4

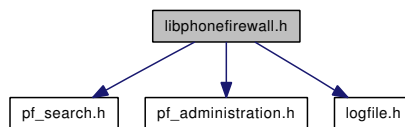
# Phone Firewall File Documentation

### 4.1 libphonefirewall.h File Reference

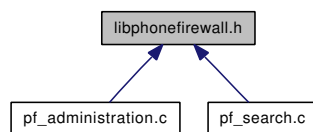
API of the phone firewall.

```
#include "pf_search.h"
#include "pf_administration.h"
#include "logfile.h"
```

Include dependency graph for libphonefirewall.h:



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



### Data Structures

- struct [Entry](#)

### Defines

- #define [PRIO\\_ALL](#) -999
- #define [DB\\_FILE](#) "db/phone-firewall.db"
- #define [STMT\\_SIZE](#) 1024
- #define [MAX\\_LINE\\_LENGTH](#) 512

- `#define TB_COUNTRYCODE "countrycode"`
- `#define TB_AREACODE "areacode"`
- `#define TB_NUMBER "number"`
- `#define TB_NAME "name"`
- `#define TB_REASON "reason"`
- `#define TB_PRIORITY "priority"`
- `#define BLACKLIST_FLAG 0x01`
- `#define WHITELIST_FLAG 0x02`
- `#define COMPONENT_NAME "phone firewall"`

### 4.1.1 Detailed Description

API of the phone firewall.

**Author:**

Alex Oberhauser

The header file of the Phone Firewall. Blocks or accepts incoming phone calls, so it's possible to prevent disturbing phone calls. Provides a API which can used by other application to build nice programs.

Definition in file [libphonefirewall.h](#).

### 4.1.2 Define Documentation

#### 4.1.2.1 `#define BLACKLIST_FLAG 0x01`

Definition at line 48 of file [libphonefirewall.h](#).

Referenced by [check\\_entry\(\)](#).

#### 4.1.2.2 `#define COMPONENT_NAME "phone firewall"`

Definition at line 51 of file [libphonefirewall.h](#).

#### 4.1.2.3 `#define DB_FILE "db/phone-firewall.db"`

Definition at line 37 of file [libphonefirewall.h](#).

Referenced by [add\\_entry\(\)](#), [change\\_name\(\)](#), [change\\_number\(\)](#), [change\\_priority\(\)](#), [change\\_reason\(\)](#), [check\\_entry\(\)](#), [get\\_entry\\_by\\_name\(\)](#), [get\\_entry\\_by\\_number\(\)](#), [get\\_entry\\_by\\_reason\(\)](#), and [rm\\_entry\(\)](#).

#### 4.1.2.4 `#define MAX_LINE_LENGTH 512`

Definition at line 39 of file [libphonefirewall.h](#).

Referenced by [add\\_entry\(\)](#), [change\\_name\(\)](#), [change\\_number\(\)](#), [change\\_priority\(\)](#), [change\\_reason\(\)](#), [check\\_entry\(\)](#), [get\\_entry\\_by\\_name\(\)](#), [get\\_entry\\_by\\_number\(\)](#), [get\\_entry\\_by\\_reason\(\)](#), and [rm\\_entry\(\)](#).

#### 4.1.2.5 #define PRIO\_ALL -999

Definition at line 36 of file libphonefirewall.h.

Referenced by add\_entry(), and evaluate\_stmt().

#### 4.1.2.6 #define STMT\_SIZE 1024

Definition at line 38 of file libphonefirewall.h.

Referenced by add\_entry(), change\_name(), change\_number(), change\_priority(), change\_reason(), check\_entry(), get\_entry\_by\_name(), get\_entry\_by\_number(), get\_entry\_by\_reason(), and rm\_entry().

#### 4.1.2.7 #define TB\_AREACODE "areacode"

Definition at line 42 of file libphonefirewall.h.

Referenced by add\_entry(), change\_name(), change\_number(), change\_priority(), change\_reason(), check\_entry(), evaluate\_stmt(), find\_entry(), get\_entry\_by\_name(), get\_entry\_by\_number(), get\_entry\_by\_reason(), and rm\_entry().

#### 4.1.2.8 #define TB\_COUNTRYCODE "countrycode"

Definition at line 41 of file libphonefirewall.h.

Referenced by add\_entry(), change\_name(), change\_number(), change\_priority(), change\_reason(), check\_entry(), evaluate\_stmt(), find\_entry(), get\_entry\_by\_name(), get\_entry\_by\_number(), get\_entry\_by\_reason(), and rm\_entry().

#### 4.1.2.9 #define TB\_NAME "name"

Definition at line 44 of file libphonefirewall.h.

Referenced by add\_entry(), change\_name(), find\_entry(), get\_entry\_by\_name(), get\_entry\_by\_number(), and get\_entry\_by\_reason().

#### 4.1.2.10 #define TB\_NUMBER "number"

Definition at line 43 of file libphonefirewall.h.

Referenced by add\_entry(), change\_name(), change\_number(), change\_priority(), change\_reason(), check\_entry(), evaluate\_stmt(), find\_entry(), get\_entry\_by\_name(), get\_entry\_by\_number(), get\_entry\_by\_reason(), and rm\_entry().

#### 4.1.2.11 #define TB\_PRIORITY "priority"

Definition at line 46 of file libphonefirewall.h.

Referenced by add\_entry(), change\_priority(), check\_entry(), evaluate\_stmt(), find\_entry(), get\_entry\_by\_name(), get\_entry\_by\_number(), and get\_entry\_by\_reason().

**4.1.2.12 #define TB\_REASON "reason"**

Definition at line 45 of file libphonefirewall.h.

Referenced by add\_entry(), change\_reason(), find\_entry(), get\_entry\_by\_name(), get\_entry\_by\_number(), and get\_entry\_by\_reason().

**4.1.2.13 #define WHITELIST\_FLAG 0x02**

Definition at line 49 of file libphonefirewall.h.

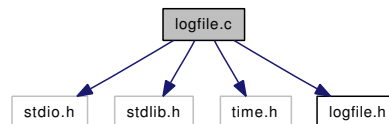
Referenced by add\_entry(), change\_name(), change\_number(), change\_priority(), change\_reason(), check\_entry(), get\_entry\_by\_name(), get\_entry\_by\_number(), get\_entry\_by\_reason(), and rm\_entry().



## 4.2 logfile.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "logfile.h"
```

Include dependency graph for logfile.c:



### Functions

- char \* [asctime](#) (const struct tm \*timeptr)
- int [write\\_logentry](#) (char \*msg, char \*component, int flag)

#### 4.2.1 Function Documentation

##### 4.2.1.1 char\* asctime (const struct tm \* timeptr)

Compounds a humand readable date and time string.

##### Parameters:

*timeptr* A pointer to the actual time.

##### Returns:

The date and time as a string.

Definition at line 33 of file logfile.c.

Referenced by write\_logentry().

Here is the caller graph for this function:



##### 4.2.1.2 int write\_logentry (char \* msg, char \* component, int flag)

Writes a logfile enty.

##### Parameters:

*msg* The message which should be written in the logfile.

***component*** The program which calls the write\_logentry function, for example "phonefirewall"

***flag*** What message should be written. Use the defined flags.

**Returns:**

-1 if something fails, otherwise 0

Definition at line 58 of file logfile.c.

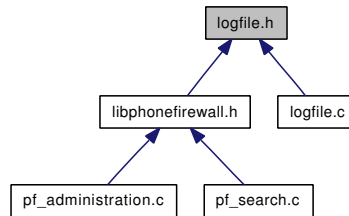
References `asctime()`, `ERR_FLAG`, `INFO_FLAG`, `LOGFILE`, `MAX_ENTRY_LENGTH`, `UNKNOWN`, and `WARN_FLAG`.

Here is the call graph for this function:



## 4.3 logfile.h File Reference

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



### Defines

- `#define LOGFILE "/tmp/phonefirewall.log"`
- `#define MAX_ENTRY_LENGTH 128`
- `#define UNKNOWN 0`
- `#define ERR_FLAG 1`
- `#define WARN_FLAG 2`
- `#define INFO_FLAG 3`
- `#define UNKNOWN_LOG(m) write_logentry(m, COMPONENT_NAME, UNKNOWN_FLAG)`
- `#define ERR_LOG(m) write_logentry(m, COMPONENT_NAME, ERR_FLAG)`
- `#define WARN_LOG(m) write_logentry(m, COMPONENT_NAME, WARN_FLAG)`
- `#define INFO_LOG(m) write_logentry(m, COMPONENT_NAME, INFO_FLAG)`

### Functions

- `int write_logentry(char *msg, char *component, int flag)`

#### 4.3.1 Define Documentation

##### 4.3.1.1 `#define ERR_FLAG 1`

Definition at line 25 of file logfile.h.

Referenced by `write_logentry()`.

##### 4.3.1.2 `#define ERR_LOG(m) write_logentry(m, COMPONENT_NAME, ERR_FLAG)`

Definition at line 30 of file logfile.h.

Referenced by `add_entry()`, `change_name()`, `change_number()`, `change_priority()`, `change_reason()`, `check_entry()`, `get_entry_by_name()`, `get_entry_by_number()`, `get_entry_by_reason()`, and `rm_entry()`.

##### 4.3.1.3 `#define INFO_FLAG 3`

Definition at line 27 of file logfile.h.

Referenced by `write_logentry()`.

**4.3.1.4 #define INFO\_LOG(m) write\_logentry(m, COMPONENT\_NAME, INFO\_FLAG)**

Definition at line 32 of file logfile.h.

Referenced by check\_entry().

**4.3.1.5 #define LOGFILE "/tmp/phonefirewall.log"**

Definition at line 21 of file logfile.h.

Referenced by write\_logentry().

**4.3.1.6 #define MAX\_ENTRY\_LENGTH 128**

Definition at line 22 of file logfile.h.

Referenced by write\_logentry().

**4.3.1.7 #define UNKNOWN 0**

Definition at line 24 of file logfile.h.

Referenced by write\_logentry().

**4.3.1.8 #define UNKNOWN\_LOG(m) write\_logentry(m, COMPONENT\_NAME, UNKNOWN\_FLAG)**

Definition at line 29 of file logfile.h.

**4.3.1.9 #define WARN\_FLAG 2**

Definition at line 26 of file logfile.h.

Referenced by write\_logentry().

**4.3.1.10 #define WARN\_LOG(m) write\_logentry(m, COMPONENT\_NAME, WARN\_FLAG)**

Definition at line 31 of file logfile.h.

**4.3.2 Function Documentation****4.3.2.1 int write\_logentry (char \* *msg*, char \* *component*, int *flag*)**

Writes a logfile entry.

**Parameters:**

*msg* The message which should be written in the logfile.

*component* The program which calls the write\_logentry function, for example "phonefirewall"

*flag* What message should be written. Use the defined flags.

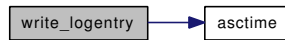
**Returns:**

-1 if something fails, otherwise 0

Definition at line 58 of file logfile.c.

References `asctime()`, `ERR_FLAG`, `INFO_FLAG`, `LOGFILE`, `MAX_ENTRY_LENGTH`, `UNKNOWN`, and `WARN_FLAG`.

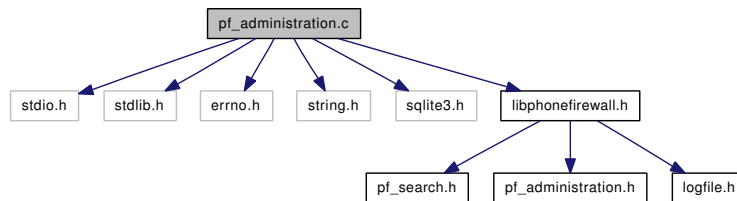
Here is the call graph for this function:



## 4.4 pf\_administration.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <sqlite3.h>
#include "libphonefirewall.h"
```

Include dependency graph for pf\_administration.c:



### Functions

- int [evaluate\\_stmt](#) (sqlite3\_stmt \*pp\_stmt, struct [Entry](#) \*p\_entry)
- int [add\\_entry](#) (int country\_code, int area\_code, unsigned long long number, char \*name, char \*reason, int priority, int listflag)
- int [rm\\_entry](#) (int country\_code, int area\_code, unsigned long long number, int listflag)
- int [check\\_entry](#) (int country\_code, int area\_code, unsigned long long number, int priority, int listflag)
- int [change\\_name](#) (int country\_code, int area\_code, unsigned long long number, char \*new\_name, int listflag)
- int [change\\_number](#) (int country\_code, int area\_code, unsigned long long number, int new\_country\_code, int new\_area\_code, unsigned long long new\_number, int listflag)
- int [change\\_reason](#) (int country\_code, int area\_code, unsigned long long number, char \*new\_reason, int listflag)
- int [change\\_priority](#) (int country\_code, int area\_code, unsigned long long number, int new\_priority, int listflag)

### 4.4.1 Function Documentation

#### 4.4.1.1 int add\_entry (int country\_code, int area\_code, unsigned long long number, char \* name, char \* reason, int priority, int listflag)

Add a number to the blacklist/whitelist. The number will be blocked after that.

##### Parameters:

- country\_code** The country code (for example 39 for Italy, 43 for Austria, and so one)
- area\_code** The area code which indicates your mobile operator.
- number** The telephone number of the person (without country and area code).
- name** The name of the person.

*reason* Why you have blocked this person.

*priority* Gives the [entry](#) a priority. 0 is standard. If the priority is higher the value will be also blocked/accepted if a higher priority is choosen.

*listflag* A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

The value "PRIO\_ALL" stands for all priorities.

#### Returns:

If all goes well 0 (zero) otherwise -1 and the errno variable will be set..

Definition at line 74 of file pf\_administration.c.

References DB\_FILE, ERR\_LOG, MAX\_LINE\_LENGTH, PRIO\_ALL, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NAME, TB\_NUMBER, TB\_PRIORITY, TB\_REASON, and WHITELIST\_FLAG.

#### 4.4.1.2 int change\_name (int country\_code, int area\_code, unsigned long long number, char \* new\_name, int listflag)

Changes the name of the [entry](#). For the unique identification you need to enter the country code, area code and the number because this tripple identifies the [entry](#).

#### Parameters:

*country\_code* The country code (for example 39 for Italy, 43 for Austria, and so one)

*area\_code* The area code which indicates your mobile operator.

*number* The telephone number of the person (without country and area code).

*new\_name* The new name

*listflag* A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

#### Returns:

If the number was changed 1, otherwise 0.

Definition at line 227 of file pf\_administration.c.

References DB\_FILE, ERR\_LOG, MAX\_LINE\_LENGTH, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NAME, TB\_NUMBER, and WHITELIST\_FLAG.

#### 4.4.1.3 int change\_number (int country\_code, int area\_code, unsigned long long number, int new\_country\_code, int new\_area\_code, unsigned long long new\_number, int listflag)

Changes the number of the [entry](#). You need to enter the country code, area code and the number because this tripple identifies the [entry](#).

#### Parameters:

*country\_code* The country code (for example 39 for Italy, 43 for Austria, and so one)

*area\_code* The area code which indicates your mobile operator.

**number** The telephone number of the person (without country and area code).

**new\_country\_code** The new country code.

**new\_area\_code** The new area code.

**new\_number** The new number.

**listflag** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

#### Returns:

If the number was changed 1, otherwise 0.

Definition at line 266 of file pf\_administration.c.

References DB\_FILE, ERR\_LOG, MAX\_LINE\_LENGTH, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NUMBER, and WHITELIST\_FLAG.

#### 4.4.1.4 int change\_priority (int country\_code, int area\_code, unsigned long long number, int new\_priority, int listflag)

Changes the priority of the [entry](#). For the unique identification you need to enter the country code, area code and the number because this tripple identifies the [entry](#).

#### Parameters:

**country\_code** The country code (for example 39 for Italy, 43 for Austria, and so one)

**area\_code** The area code which indicates your mobile operator.

**number** The telephone number of the person (without country and area code).

**new\_priority** The new priority.

**listflag** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

#### Returns:

If the number was changed 1, otherwise 0.

Definition at line 346 of file pf\_administration.c.

References DB\_FILE, ERR\_LOG, MAX\_LINE\_LENGTH, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NUMBER, TB\_PRIORITY, and WHITELIST\_FLAG.

#### 4.4.1.5 int change\_reason (int country\_code, int area\_code, unsigned long long number, char \* new\_reason, int listflag)

Changes the reason of the [entry](#). For the unique identification you need to enter the country code, area code and the number because this tripple identifies the [entry](#).

#### Parameters:

**country\_code** The country code (for example 39 for Italy, 43 for Austria, and so one)

**area\_code** The area code which indicates your mobile operator.

**number** The telephone number of the person (without country and area code).



***new\_reason*** The new reason.

***listflag*** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

#### Returns:

If the number was changed 1, otherwise 0.

Definition at line 307 of file pf\_administration.c.

References DB\_FILE, ERR\_LOG, MAX\_LINE\_LENGTH, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NUMBER, TB\_REASON, and WHITELIST\_FLAG.

#### 4.4.1.6 int check\_entry (int country\_code, int area\_code, unsigned long long number, int priority, int listflag)

Checks if a number is on the blacklist/whitelist.

#### Parameters:

***country\_code*** The country code (for example 39 for Italy, 43 for Austria, and so one)

***area\_code*** The area code which indicates your mobile operator.

***number*** The telephone number of the person (without country and area code).

***priority*** Gives the [entry](#) a priority. 0 is standard. If the priority is higher the value will be also blocked/accepted if a higher priority is choosen.

***listflag*** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

The value "PRIO\_ALL" stands for all priorities.

#### Returns:

If the number was found 1, otherwise 0.

Definition at line 155 of file pf\_administration.c.

References Entry::area\_code, BLACKLIST\_FLAG, Entry::country\_code, DB\_FILE, ERR\_LOG, evaluate\_stmt(), INFO\_LOG, MAX\_LINE\_LENGTH, Entry::number, Entry::priority, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NUMBER, TB\_PRIORITY, and WHITELIST\_FLAG.

Here is the call graph for this function:



#### 4.4.1.7 int evaluate\_stmt (sqlite3\_stmt \* pp\_stmt, struct Entry \* p\_entry)

Definition at line 27 of file pf\_administration.c.

References Entry::area\_code, Entry::country\_code, Entry::number, PRIO\_ALL, Entry::priority, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NUMBER, and TB\_PRIORITY.

Referenced by check\_entry().

Here is the caller graph for this function:



#### 4.4.1.8 `int rm_entry (int country_code, int area_code, unsigned long long number, int listflag)`

Removes a number from the blacklist/whitelist.

##### Parameters:

*country\_code* The country code (for example 39 for Italy, 43 for Austria, and so one)

*area\_code* The area code which indicates your mobile operator.

*number* The number which will be deleted.

*listflag* A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

##### Returns:

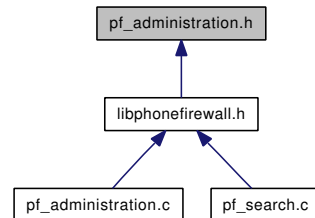
If all goes right 0, otherwise an error code.

Definition at line 119 of file `pf_administration.c`.

References `DB_FILE`, `ERR_LOG`, `MAX_LINE_LENGTH`, `STMT_SIZE`, `TB_AREACODE`, `TB_COUNTRYCODE`, `TB_NUMBER`, and `WHITELIST_FLAG`.

## 4.5 pf\_administration.h File Reference

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



### Functions

- int [add\\_entry](#) (int country\_code, int area\_code, unsigned long long number, char \*name, char \*reason, int priority, int listflag)
- int [rm\\_entry](#) (int country\_code, int area\_code, unsigned long long number, int listflag)
- int [check\\_entry](#) (int country\_code, int area\_code, unsigned long long number, int priority, int listflag)
- int [change\\_name](#) (int country\_code, int area\_code, unsigned long long number, char \*new\_name, int listflag)
- int [change\\_number](#) (int country\_code, int area\_code, unsigned long long number, int new\_country\_code, int new\_area\_code, unsigned long long new\_number, int listflag)
- int [change\\_reason](#) (int country\_code, int area\_code, unsigned long long number, char \*new\_reason, int listflag)
- int [change\\_priority](#) (int country\_code, int area\_code, unsigned long long number, int new\_priority, int listflag)

### 4.5.1 Function Documentation

#### 4.5.1.1 int add\_entry (int country\_code, int area\_code, unsigned long long number, char \* name, char \* reason, int priority, int listflag)

Add a number to the blacklist/whitelist. The number will be blocked after that.

##### Parameters:

**country\_code** The country code (for example 39 for Italy, 43 for Austria, and so one)

**area\_code** The area code which indicates your mobile operator.

**number** The telephone number of the person (without country and area code).

**name** The name of the person.

**reason** Why you have blocked this person.

**priority** Gives the [entry](#) a priority. 0 is standard. If the priority is higher the value will be also blocked/accepted if a higher priority is choosen.

**listflag** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

The value "PRIO\_ALL" stands for all priorities.

**Returns:**

If all goes well 0 (zero) otherwise -1 and the errno variable will be set..

Definition at line 74 of file pf\_administration.c.

References DB\_FILE, ERR\_LOG, MAX\_LINE\_LENGTH, PRIO\_ALL, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NAME, TB\_NUMBER, TB\_PRIORITY, TB\_REASON, and WHITELIST\_FLAG.

**4.5.1.2 int change\_name (int country\_code, int area\_code, unsigned long long number, char \* new\_name, int listflag)**

Changes the name of the [entry](#). For the unique identification you need to enter the country code, area code and the number because this tripple identifies the [entry](#).

**Parameters:**

**country\_code** The country code (for example 39 for Italy, 43 for Austria, and so one)

**area\_code** The area code which indicates your mobile operator.

**number** The telephone number of the person (without country and area code).

**new\_name** The new name

**listflag** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

**Returns:**

If the number was changed 1, otherwise 0.

Definition at line 227 of file pf\_administration.c.

References DB\_FILE, ERR\_LOG, MAX\_LINE\_LENGTH, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NAME, TB\_NUMBER, and WHITELIST\_FLAG.

**4.5.1.3 int change\_number (int country\_code, int area\_code, unsigned long long number, int new\_country\_code, int new\_area\_code, unsigned long long new\_number, int listflag)**

Changes the number of the [entry](#). You need to enter the country code, area code and the number because this tripple identifies the [entry](#).

**Parameters:**

**country\_code** The country code (for example 39 for Italy, 43 for Austria, and so one)

**area\_code** The area code which indicates your mobile operator.

**number** The telephone number of the person (without country and area code).

**new\_country\_code** The new country code.

**new\_area\_code** The new area code.

**new\_number** The new number.

**listflag** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

**Returns:**

If the number was changed 1, otherwise 0.

Definition at line 266 of file pf\_administration.c.

References DB\_FILE, ERR\_LOG, MAX\_LINE\_LENGTH, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NUMBER, and WHITELIST\_FLAG.

**4.5.1.4 int change\_priority (int country\_code, int area\_code, unsigned long long number, int new\_priority, int listflag)**

Changes the priority of the [entry](#). For the unique identification you need to enter the country code, area code and the number because this tripple identifies the [entry](#).

**Parameters:**

**country\_code** The country code (for example 39 for Italy, 43 for Austria, and so one)

**area\_code** The area code which indicates your mobile operator.

**number** The telephone number of the person (without country and area code).

**new\_priority** The new priority.

**listflag** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

**Returns:**

If the number was changed 1, otherwise 0.

Definition at line 346 of file pf\_administration.c.

References DB\_FILE, ERR\_LOG, MAX\_LINE\_LENGTH, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NUMBER, TB\_PRIORITY, and WHITELIST\_FLAG.

**4.5.1.5 int change\_reason (int country\_code, int area\_code, unsigned long long number, char \* new\_reason, int listflag)**

Changes the reason of the [entry](#). For the unique identification you need to enter the country code, area code and the number because this tripple identifies the [entry](#).

**Parameters:**

**country\_code** The country code (for example 39 for Italy, 43 for Austria, and so one)

**area\_code** The area code which indicates your mobile operator.

**number** The telephone number of the person (without country and area code).

**new\_reason** The new reason.

**listflag** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

**Returns:**

If the number was changed 1, otherwise 0.

Definition at line 307 of file pf\_administration.c.

References DB\_FILE, ERR\_LOG, MAX\_LINE\_LENGTH, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NUMBER, TB\_REASON, and WHITELIST\_FLAG.

#### 4.5.1.6 `int check_entry (int country_code, int area_code, unsigned long long number, int priority, int listflag)`

Checks if a number is on the blacklist/whitelist.

##### Parameters:

**country\_code** The country code (for example 39 for Italy, 43 for Austria, and so one)

**area\_code** The area code which indicates your mobile operator.

**number** The telephone number of the person (without country and area code).

**priority** Gives the [entry](#) a priority. 0 is standard. If the priority is higher the value will be also blocked/accepted if a higher priority is choosen.

**listflag** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

The value "PRIO\_ALL" stands for all priorities.

##### Returns:

If the number was found 1, otherwise 0.

Definition at line 155 of file pf\_administration.c.

References `Entry::area_code`, `BLACKLIST_FLAG`, `Entry::country_code`, `DB_FILE`, `ERR_LOG`, `evaluate_stmt()`, `INFO_LOG`, `MAX_LINE_LENGTH`, `Entry::number`, `Entry::priority`, `STMT_SIZE`, `TB_AREACODE`, `TB_COUNTRYCODE`, `TB_NUMBER`, `TB_PRIORITY`, and `WHITELIST_FLAG`.

Here is the call graph for this function:



#### 4.5.1.7 `int rm_entry (int country_code, int area_code, unsigned long long number, int listflag)`

Removes a number from the blacklist/whitelist.

##### Parameters:

**country\_code** The country code (for example 39 for Italy, 43 for Austria, and so one)

**area\_code** The area code which indicates your mobile operator.

**number** The number which will be deleted.

**listflag** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

##### Returns:

If all goes right 0, otherwise an error code.

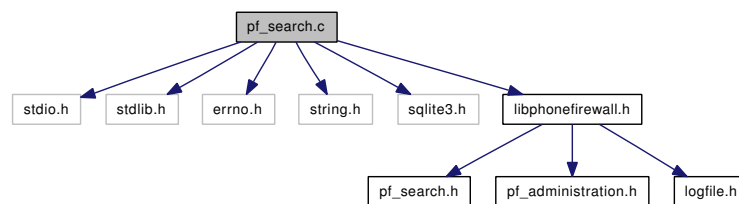
Definition at line 119 of file pf\_administration.c.

References `DB_FILE`, `ERR_LOG`, `MAX_LINE_LENGTH`, `STMT_SIZE`, `TB_AREACODE`, `TB_COUNTRYCODE`, `TB_NUMBER`, and `WHITELIST_FLAG`.

## 4.6 pf\_search.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <sqlite3.h>
#include "libphonefirewall.h"
```

Include dependency graph for pf\_search.c:



### Defines

- #define [ASCII\\_PERCENT\\_CHAR](#) 37

### Functions

- struct [Entry](#) \* [insert\\_into\\_list](#) (struct [Entry](#) \*p\_root, struct [Entry](#) \*p\_entry)
- struct [Entry](#) \* [find\\_entry](#) (sqlite3\_stmt \*pp\_stmt)
- struct [Entry](#) \* [get\\_entry\\_by\\_name](#) (char \*name, int listflag)
- struct [Entry](#) \* [get\\_entry\\_by\\_number](#) (int country\_code, int area\_code, unsigned long long number, int listflag)
- struct [Entry](#) \* [get\\_entry\\_by\\_reason](#) (char \*reason, int listflag)

#### 4.6.1 Define Documentation

##### 4.6.1.1 #define ASCII\_PERCENT\_CHAR 37

Definition at line 27 of file pf\_search.c.

Referenced by [get\\_entry\\_by\\_name\(\)](#), [get\\_entry\\_by\\_number\(\)](#), and [get\\_entry\\_by\\_reason\(\)](#).

#### 4.6.2 Function Documentation

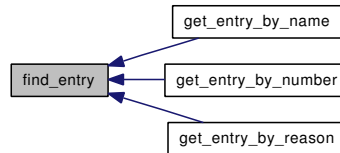
##### 4.6.2.1 struct Entry\* find\_entry (sqlite3\_stmt \* pp\_stmt) [read]

Definition at line 76 of file pf\_search.c.

References [Entry::area\\_code](#), [Entry::country\\_code](#), [Entry::name](#), [Entry::number](#), [Entry::priority](#), [Entry::reason](#), [TB\\_AREACODE](#), [TB\\_COUNTRYCODE](#), [TB\\_NAME](#), [TB\\_NUMBER](#), [TB\\_PRIORITY](#), and [TB\\_REASON](#).

Referenced by `get_entry_by_name()`, `get_entry_by_number()`, and `get_entry_by_reason()`.

Here is the caller graph for this function:



#### 4.6.2.2 `struct Entry* get_entry_by_name (char * name, int listflag)` [read]

Search a [entry](#) by name.

##### Parameters:

***name*** The name of the person which is blocked.

***listflag*** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

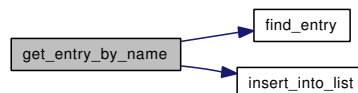
##### Returns:

[entry](#) Returns the entries which are found in a linked list.

Definition at line 112 of file `pf_search.c`.

References `ASCII_PERCENT_CHAR`, `DB_FILE`, `ERR_LOG`, `find_entry()`, `insert_into_list()`, `MAX_LINE_LENGTH`, `STMT_SIZE`, `TB_AREACODE`, `TB_COUNTRYCODE`, `TB_NAME`, `TB_NUMBER`, `TB_PRIORITY`, `TB_REASON`, and `WHITELIST_FLAG`.

Here is the call graph for this function:



#### 4.6.2.3 `struct Entry* get_entry_by_number (int country_code, int area_code, unsigned long long number, int listflag)` [read]

Search a entries by number (country code + area code + number).

##### Parameters:

***country\_code*** The country code (for example 39 for Italy, 43 for Austria, and so one)

***area\_code*** The area code which indicates your mobile operator.

***number*** The telephone number of the person (without country and area code).

***listflag*** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).



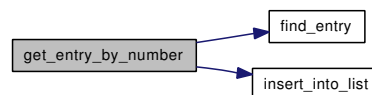
**Returns:**

[entry](#) Returns the entries which are found in a linked list.

Definition at line 158 of file pf\_search.c.

References ASCII\_PERCENT\_CHAR, DB\_FILE, ERR\_LOG, find\_entry(), insert\_into\_list(), MAX\_LINE\_LENGTH, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NAME, TB\_NUMBER, TB\_PRIORITY, TB\_REASON, and WHITELIST\_FLAG.

Here is the call graph for this function:



#### 4.6.2.4 struct Entry\* get\_entry\_by\_reason(char \*reason, int listflag) [read]

Search a entries by reason.

**Parameters:**

**reason** The reason why a person is blocked/accpeted.

**listflag** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

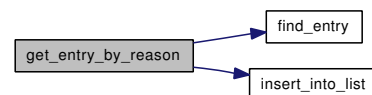
**Returns:**

[entry](#) Returns the entries which are found in a linked list.

Definition at line 206 of file pf\_search.c.

References ASCII\_PERCENT\_CHAR, DB\_FILE, ERR\_LOG, find\_entry(), insert\_into\_list(), MAX\_LINE\_LENGTH, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NAME, TB\_NUMBER, TB\_PRIORITY, TB\_REASON, and WHITELIST\_FLAG.

Here is the call graph for this function:



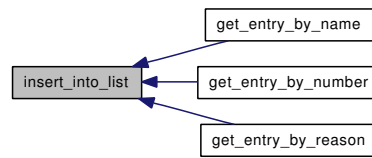
#### 4.6.2.5 struct Entry\* insert\_into\_list(struct Entry \*p\_root, struct Entry \*p\_entry) [read]

Definition at line 29 of file pf\_search.c.

References Entry::area\_code, Entry::country\_code, Entry::name, Entry::next, Entry::number, and Entry::reason.

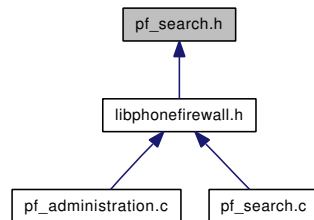
Referenced by get\_entry\_by\_name(), get\_entry\_by\_number(), and get\_entry\_by\_reason().

Here is the caller graph for this function:



## 4.7 pf\_search.h File Reference

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



### Functions

- struct [Entry](#) \* [get\\_entry\\_by\\_name](#) (char \*name, int listflag)
- struct [Entry](#) \* [get\\_entry\\_by\\_number](#) (int country\_code, int area\_code, unsigned long long number, int listflag)
- struct [Entry](#) \* [get\\_entry\\_by\\_reason](#) (char \*reason, int listflag)

#### 4.7.1 Function Documentation

##### 4.7.1.1 struct Entry\* [get\\_entry\\_by\\_name](#) (char \* *name*, int *listflag*) [read]

Search a [entry](#) by name.

##### Parameters:

***name*** The name of the person which is blocked.

***listflag*** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

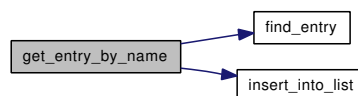
##### Returns:

[entry](#) Returns the entries which are found in a linked list.

Definition at line 112 of file pf\_search.c.

References [ASCII\\_PERCENT\\_CHAR](#), [DB\\_FILE](#), [ERR\\_LOG](#), [find\\_entry\(\)](#), [insert\\_into\\_list\(\)](#), [MAX\\_LINE\\_LENGTH](#), [STMT\\_SIZE](#), [TB\\_AREACODE](#), [TB\\_COUNTRYCODE](#), [TB\\_NAME](#), [TB\\_NUMBER](#), [TB\\_PRIORITY](#), [TB\\_REASON](#), and [WHITELIST\\_FLAG](#).

Here is the call graph for this function:



#### 4.7.1.2 **struct Entry\* get\_entry\_by\_number (int *country\_code*, int *area\_code*, unsigned long long *number*, int *listflag*)** [read]

Search a entries by number (country code + area code + number).

##### Parameters:

***country\_code*** The country code (for example 39 for Italy, 43 for Austria, and so one)

***area\_code*** The area code which indicates your mobile operator.

***number*** The telephone number of the person (without country and area code).

***listflag*** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

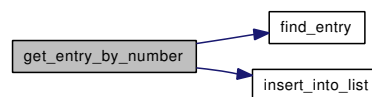
##### Returns:

[entry](#) Returns the entries which are found in a linked list.

Definition at line 158 of file pf\_search.c.

References ASCII\_PERCENT\_CHAR, DB\_FILE, ERR\_LOG, find\_entry(), insert\_into\_list(), MAX\_LINE\_LENGTH, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NAME, TB\_NUMBER, TB\_PRIORITY, TB\_REASON, and WHITELIST\_FLAG.

Here is the call graph for this function:



#### 4.7.1.3 **struct Entry\* get\_entry\_by\_reason (char \* *reason*, int *listflag*)** [read]

Search a entries by reason.

##### Parameters:

***reason*** The reason why a person is blocked/accpeted.

***listflag*** A flag, which indicates if you would use the blacklist (BLACKLIST\_FLAG) or the whitelist (WHITELIST\_FLAG).

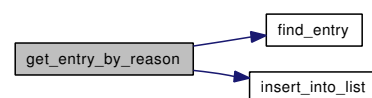
##### Returns:

[entry](#) Returns the entries which are found in a linked list.

Definition at line 206 of file pf\_search.c.

References ASCII\_PERCENT\_CHAR, DB\_FILE, ERR\_LOG, find\_entry(), insert\_into\_list(), MAX\_LINE\_LENGTH, STMT\_SIZE, TB\_AREACODE, TB\_COUNTRYCODE, TB\_NAME, TB\_NUMBER, TB\_PRIORITY, TB\_REASON, and WHITELIST\_FLAG.

Here is the call graph for this function:



# Index

- add\_entry
  - pf\_administration.c, [18](#)
  - pf\_administration.h, [23](#)
- area\_code
  - Entry, [5](#)
- ASCII\_PERCENT\_CHAR
  - pf\_search.c, [27](#)
- asctime
  - logfile.c, [13](#)
- BLACKLIST\_FLAG
  - libphonefirewall.h, [10](#)
- change\_name
  - pf\_administration.c, [19](#)
  - pf\_administration.h, [24](#)
- change\_number
  - pf\_administration.c, [19](#)
  - pf\_administration.h, [24](#)
- change\_priority
  - pf\_administration.c, [20](#)
  - pf\_administration.h, [25](#)
- change\_reason
  - pf\_administration.c, [20](#)
  - pf\_administration.h, [25](#)
- check\_entry
  - pf\_administration.c, [21](#)
  - pf\_administration.h, [25](#)
- COMPONENT\_NAME
  - libphonefirewall.h, [10](#)
- country\_code
  - Entry, [5](#)
- DB\_FILE
  - libphonefirewall.h, [10](#)
- Entry, [5](#)
  - area\_code, [5](#)
  - country\_code, [5](#)
  - name, [6](#)
  - next, [6](#)
  - number, [6](#)
  - priority, [6](#)
  - reason, [6](#)
- entry, [7](#)
- ERR\_FLAG
  - logfile.h, [15](#)
- ERR\_LOG
  - logfile.h, [15](#)
- evaluate\_stmt
  - pf\_administration.c, [21](#)
- find\_entry
  - pf\_search.c, [27](#)
- get\_entry\_by\_name
  - pf\_search.c, [28](#)
  - pf\_search.h, [31](#)
- get\_entry\_by\_number
  - pf\_search.c, [28](#)
  - pf\_search.h, [31](#)
- get\_entry\_by\_reason
  - pf\_search.c, [29](#)
  - pf\_search.h, [32](#)
- INFO\_FLAG
  - logfile.h, [15](#)
- INFO\_LOG
  - logfile.h, [15](#)
- insert\_into\_list
  - pf\_search.c, [29](#)
- libphonefirewall.h, [9](#)
  - BLACKLIST\_FLAG, [10](#)
  - COMPONENT\_NAME, [10](#)
  - DB\_FILE, [10](#)
  - MAX\_LINE\_LENGTH, [10](#)
  - PRIO\_ALL, [10](#)
  - STMT\_SIZE, [11](#)
  - TB\_AREACODE, [11](#)
  - TB\_COUNTRYCODE, [11](#)
  - TB\_NAME, [11](#)
  - TB\_NUMBER, [11](#)
  - TB\_PRIORITY, [11](#)
  - TB\_REASON, [11](#)
  - WHITELIST\_FLAG, [12](#)
- LOGFILE
  - logfile.h, [16](#)
- logfile.c, [13](#)
  - asctime, [13](#)
  - write\_logentry, [13](#)
- logfile.h, [15](#)

- ERR\_FLAG, 15
- ERR\_LOG, 15
- INFO\_FLAG, 15
- INFO\_LOG, 15
- LOGFILE, 16
- MAX\_ENTRY\_LENGTH, 16
- UNKNOWN, 16
- UNKNOWN\_LOG, 16
- WARN\_FLAG, 16
- WARN\_LOG, 16
- write\_logentry, 16
- MAX\_ENTRY\_LENGTH
  - logfile.h, 16
- MAX\_LINE\_LENGTH
  - libphonefirewall.h, 10
- name
  - Entry, 6
- next
  - Entry, 6
- number
  - Entry, 6
- pf\_administration.c, 18
  - add\_entry, 18
  - change\_name, 19
  - change\_number, 19
  - change\_priority, 20
  - change\_reason, 20
  - check\_entry, 21
  - evaluate\_stmt, 21
  - rm\_entry, 22
- pf\_administration.h, 23
  - add\_entry, 23
  - change\_name, 24
  - change\_number, 24
  - change\_priority, 25
  - change\_reason, 25
  - check\_entry, 25
  - rm\_entry, 26
- pf\_search.c, 27
  - ASCII\_PERCENT\_CHAR, 27
  - find\_entry, 27
  - get\_entry\_by\_name, 28
  - get\_entry\_by\_number, 28
  - get\_entry\_by\_reason, 29
  - insert\_into\_list, 29
- pf\_search.h, 31
  - get\_entry\_by\_name, 31
  - get\_entry\_by\_number, 31
  - get\_entry\_by\_reason, 32
- PRIO\_ALL
  - libphonefirewall.h, 10
- priority
  - Entry, 6
- reason
  - Entry, 6
- rm\_entry
  - pf\_administration.c, 22
  - pf\_administration.h, 26
- STMT\_SIZE
  - libphonefirewall.h, 11
- TB\_AREACODE
  - libphonefirewall.h, 11
- TB\_COUNTRYCODE
  - libphonefirewall.h, 11
- TB\_NAME
  - libphonefirewall.h, 11
- TB\_NUMBER
  - libphonefirewall.h, 11
- TB\_PRIORITY
  - libphonefirewall.h, 11
- TB\_REASON
  - libphonefirewall.h, 11
- UNKNOWN
  - logfile.h, 16
- UNKNOWN\_LOG
  - logfile.h, 16
- WARN\_FLAG
  - logfile.h, 16
- WARN\_LOG
  - logfile.h, 16
- WHITELIST\_FLAG
  - libphonefirewall.h, 12
- write\_logentry
  - logfile.c, 13
  - logfile.h, 16