Phone Firewall Reference Manual

v0.01

Generated by Doxygen 1.5.4

Thu May 8 08:02:49 2008

Contents

Chapter 1

Phone Firewall Data Structure Index

1.1 Phone Firewall Data Structures

Here are the data structures with brief descriptions:

Chapter 2

Phone Firewall File Index

2.1 Phone Firewall File List

Here is a list of all files with brief descriptions:

Chapter 3

Phone Firewall Data Structure Documentation

3.1 blacklist Struct Reference

Contains the blocked numbers.

`#include <libphonefirewall.h>`

Data Fields

- char * name
- char number [TELNR_MAXLEN+1]
- int priority
- char * reason

3.1.1 Detailed Description

Contains the blocked numbers.

Definition at line 42 of file libphonefirewall.h.

3.1.2 Field Documentation

3.1.2.1 char* blacklist::name

Definition at line 43 of file libphonefirewall.h.

3.1.2.2 char blacklist::number[TELNR_MAXLEN+1]

Definition at line 44 of file libphonefirewall.h.

3.1.2.3 int blacklist::priority

Definition at line 45 of file libphonefirewall.h.

3.1.2.4 char* blacklist::reason

Definition at line 46 of file libphonefirewall.h.

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

- libphonefirewall.h

3.2 whitelist Struct Reference

Contains the accepted numbers.

`#include <libphonefirewall.h>`

Data Fields

- char * name
- char number [TELNR_MAXLEN+1]
- int priority
- char * reason

3.2.1 Detailed Description

Contains the accepted numbers.

Definition at line 53 of file libphonefirewall.h.

3.2.2 Field Documentation

3.2.2.1 char* whitelist::name

Definition at line 54 of file libphonefirewall.h.

3.2.2.2 char whitelist::number[TELNR_MAXLEN+1]

Definition at line 55 of file libphonefirewall.h.

3.2.2.3 int whitelist::priority

Definition at line 56 of file libphonefirewall.h.

3.2.2.4 char* whitelist::reason

Definition at line 57 of file libphonefirewall.h.

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

- libphonefirewall.h

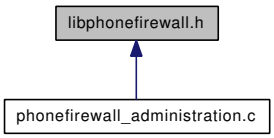
Chapter 4

Phone Firewall File Documentation

4.1 libphonefirewall.h File Reference

API of the phone firewall.

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



Data Structures

- struct blacklist

  Contains the blocked numbers.

- struct whitelist

  Contains the accepted numbers.

Defines

- #define TELNR_MAXLEN 32

Functions

- int add_blacklist_entry (char number[ ], char ∗name, char ∗reason, int priority)
- int rm_blacklist_entry (char number[ ])
- int add_whitelist_entry (char number[ ], char ∗name, char ∗reason, int priority)
- int rm_whitelist_entry (char number[ ])

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.

Implemented for the OpenMoko framework.

Definition in file libphonefirewall.h.

4.1.2 Define Documentation

4.1.2.1 #define TELNR_MAXLEN 32

The maximum length of a telephone number.

Definition at line 36 of file libphonefirewall.h.

Referenced by add_blacklist_entry().

4.1.3 Function Documentation

4.1.3.1 int add_blacklist_entry (char *number*[ ], char * *name*, char * *reason*, int *priority*)

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

Parameters:

number The telephone number of the person.

name The name of the person.

reason Why you have blocked this person.

priority Has no affect at the moment. Later one it will be possible to give each number priority. So you have more control when a number will be blocked/accepted.

Returns:

If all goes well 0 (zero) otherwise an errno code.

Definition at line 24 of file phonefirewall_administration.c.

References TELNR_MAXLEN.

4.1.3.2 int add_whitelist_entry (char *number*[ ], char * *name*, char * *reason*, int *priority*)

Add a number to the whitelist. The number will be accepted after that.

Parameters:

number The telephone number of the person.

name The name of the person.

reason Why you have blocked this person.

priority Has no affect at the moment. Later one it will be possible to give each number priority. So you have more control when a number will be blocked/accepted.

Returns:

If all goes well 0 (zero) otherwise an errno code.

Definition at line 33 of file phonefirewall_administration.c.

4.1.3.3 int rm_blacklist_entry (char *number*[ ])

Removes a blocked number from the blacklist.

Definition at line 29 of file phonefirewall_administration.c.

4.1.3.4 int rm_whitelist_entry (char *number*[ ])

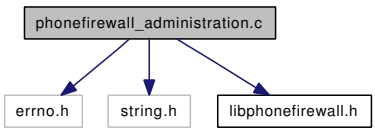
Removes a accepted number from the whitelist.

Definition at line 37 of file phonefirewall_administration.c.

4.2 phonefirewall_administration.c File Reference

```
#include <errno.h>
#include <string.h>
#include "libphonefirewall.h"
```

Include dependency graph for phonefirewall_administration.c:



Functions

- int add_blacklist_entry (char number[ ], char *name, char *reason, int priority)
- int rm_blacklist_entry (char number[ ])
- int add_whitelist_entry (char number[ ], char *name, char *reason, int priority)
- int rm_whitelist_entry (char number[ ])

4.2.1 Function Documentation

4.2.1.1 int add_blacklist_entry (char *number*[ ], char * *name*, char * *reason*, int *priority*)

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

Parameters:

number The telephone number of the person.

name The name of the person.

reason Why you have blocked this person.

priority Has no affect at the moment. Later one it will be possible to give each number priority. So you have more control when a number will be blocked/accepted.

Returns:

If all goes well 0 (zero) otherwise an errno code.

Definition at line 24 of file phonefirewall_administration.c.

References TELNR_MAXLEN.

4.2.1.2 int add_whitelist_entry (char *number*[ ], char * *name*, char * *reason*, int *priority*)

Add a number to the whitelist. The number will be accepted after that.

Parameters:

number The telephone number of the person.

name The name of the person.

reason Why you have blocked this person.

priority Has no affect at the moment. Later one it will be possible to give each number priority. So you have more control when a number will be blocked/accepted.

Returns:

If all goes well 0 (zero) otherwise an errno code.

Definition at line 33 of file phonefirewall_administration.c.

4.2.1.3 int rm_blacklist_entry (char *number*[ ])

Removes a blocked number from the blacklist.

Definition at line 29 of file phonefirewall_administration.c.

4.2.1.4 int rm_whitelist_entry (char *number*[ ])

Removes a accepted number from the whitelist.

Definition at line 37 of file phonefirewall_administration.c.

Index