

# usb-gui

## 0.1

Generated by Doxygen 1.6.3

Sun Jan 27 17:03:25 2013



# Contents

<b>1</b>	<b>Data Structure Index</b>	<b>1</b>
1.1	Data Structures . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Data Structure Documentation</b>	<b>5</b>
3.1	__request Struct Reference . . . . .	5
3.2	DEV Struct Reference . . . . .	6
3.2.1	Detailed Description . . . . .	6
<b>4</b>	<b>File Documentation</b>	<b>7</b>
4.1	/media/4ec1cddc-24fd-4aaa-a1b9-43731b032863/home/editoblez/workspace/usb/src/common/common.h File Reference . . . . .	7
4.1.1	Detailed Description . . . . .	8
4.1.2	Function Documentation . . . . .	8
4.1.2.1	get_date . . . . .	8
4.1.2.2	getmsg . . . . .	9
4.1.2.3	getuser . . . . .	9
4.1.2.4	register_log . . . . .	9
4.2	usb-gui.h File Reference . . . . .	10
4.2.1	Detailed Description . . . . .	11
4.2.2	Enumeration Type Documentation . . . . .	11
4.2.2.1	"@0 . . . . .	11
4.2.3	Function Documentation . . . . .	11
4.2.3.1	add_item . . . . .	11
4.2.3.2	authorize_device . . . . .	11
4.2.3.3	do_list_store . . . . .	12
4.2.3.4	fatal_error_signal . . . . .	12

4.2.3.5	<a href="#">get_new_device_id</a>	12
4.2.3.6	<a href="#">main</a>	12
4.2.3.7	<a href="#">read_devices_list</a>	12
4.2.3.8	<a href="#">remove_item</a>	13
4.2.3.9	<a href="#">update_tool_bar</a>	13

# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">__request</a> . . . . .	<a href="#">5</a>
<a href="#">DEV</a> (A device ) . . . . .	<a href="#">6</a>



# Chapter 2

## File Index

### 2.1 File List

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

<a href="#">/media/4ec1cddc-24fd-4aaa-a1b9-43731b032863/home/editablez/workspace/usb/src/common/common.h</a>	
<a href="#">usb-gui.h</a>	10





## Chapter 3

# Data Structure Documentation

### 3.1 \_\_request Struct Reference

#### Data Fields

- gchar \* [user](#)  
*User which connected the device.*
- const gchar \* [serial](#)  
*Serial of the device.*
- const gchar \* [devtype](#)  
*device's devtype*
- gboolean [authorized](#)  
*either is authorized or no.*
- gchar \* [date](#)  
*current date of the system.*

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

- </media/4ec1cddc-24fd-4aaa-a1b9-43731b032863/home/editoblez/workspace/usb/src/common/common.h>

## 3.2 DEV Struct Reference

A device.

```
#include <usb-gui.h>
```

### Data Fields

- const gchar \* [identifier](#)  
*name that it identify*
- const gchar \* [serial](#)  
*device's serial number*

### 3.2.1 Detailed Description

A device.

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

- [usb-gui.h](#)

## Chapter 4

# File Documentation

### 4.1 /media/4ec1cddc-24fd-4aaa-a1b9-43731b032863/home/editablez/workspace/us File Reference

```
#include <unistd.h>
#include <glib.h>
#include <glib/gprintf.h>
#include <time.h>
#include <utmp.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <syslog.h>
#include <sys/types.h>
#include <err.h>
#include <locale.h>
```

#### Data Structures

- struct [\\_\\_request](#)

#### Defines

- #define [DB](#) "/etc/usb-control/database.db"  
*Place where are the database.*
- #define [LOG](#) "/var/log/usb-control.log"  
*Place where are the application's logs.*
- #define [NO\\_AUTORIZADO](#) "No autorizado"

*Default identifier for connected device new to system.*

- #define `ENV_SERIAL` "ID\_SERIAL\_SHORT"  
*Serial number short imported from udev environment.*
- #define `ENV_DEVTYPE` "DEVTYPE"  
*Devtype imported from udev environment.*
- #define `LOCK` "/tmp/usbsecurity.lock"  
*This file avoid instances multiple of this application.*

## Typedefs

- typedef struct `__request` `request`

## Functions

- void `register_log` ()  
*Add a entry to log file.*
- gchar \* `get_date` ()  
*Get a date in gchar \* form.*
- gchar \* `getuser` ()  
*Get the current user in the system.*
- const gchar \* `getmsg` ()  
*Name of action.*

## Variables

- `request` info

### 4.1.1 Detailed Description

#### Author

Edilberto Blez D.

### 4.1.2 Function Documentation

#### 4.1.2.1 gchar\* get\_date ()

Get a date in gchar \* form.

This function is used to retrieve the date from the system.

gchar \* with the current date.

#### 4.1.2.2 const gchar\* getmsg ()

Name of action.

Get a name of the action according with the value of info.authorized.

##### Returns

a gchar \* associate.

#### 4.1.2.3 gchar\* getuser ()

Get the current user in the system.

Tries to get the user that is current connected to system. As can be connectar more than one user, the application tries to get one if this can, otherwise unknown is returned.

##### Returns

One current user o unknown if can't.

#### 4.1.2.4 void register\_log ()

Add a entry to log file.

Depend of a action this function add a new entry to a log file.

##### Parameters

*info* contain the all necessary information about the action.

## 4.2 usb-gui.h File Reference

```
#include <gtk/gtk.h>
#include "../common/common.h"
```

### Data Structures

- struct [DEV](#)  
*A device.*

### Enumerations

- enum { [COLUMN\\_MARKED](#), [COLUMN\\_ID](#), [COLUMN\\_SERIAL](#), [NUM\\_COLUMNS](#) }  
*field for the GtkListStore*

### Functions

- int [main](#) (int argc, char \*\*argv)  
*main function.*
- void [update\\_tool\\_bar](#) ()  
*Modify status of toolbar.*
- GtkWidget \* [do\\_list\\_store](#) ()  
*Function that creates the main window.*
- [DEV](#) \* [read\\_devices\\_list](#) ()  
*Create a list of [DEV](#) \* Read the database normaly in /database.db and add to [DEV](#) \* list the device there are.*
- void [remove\\_item](#) (GtkWidget \*widget, GdkEvent \*event, gpointer data)  
*Delete rows from treeview.*
- void [authorize\\_device](#) ([DEV](#) \*dev)  
*Add a device to a database.*
- void [fatal\\_error\\_signal](#) (int sig)  
*handler for exceptions.*
- void [add\\_item](#) (GtkWidget \*widget, GdkEvent \*event, gpointer data)  
*Add a item to treeview.*
- gchar \* [get\\_new\\_device\\_id](#) ()  
*Get a identify new for a row.*

## Variables

- volatile sig\_atomic\_t [fatal\\_error\\_in\\_progress](#) = 0  
*Used in signal manage.*

### 4.2.1 Detailed Description

#### Author

Edilberto Blez D.  
Abel Alfonso F.D.

### 4.2.2 Enumeration Type Documentation

#### 4.2.2.1 anonymous enum

field for the GtkListStore

#### Enumerator:

- COLUMN\_MARKED** indicate if is marked (Value = 0)  
**COLUMN\_ID** identifier (Value = 1)  
**COLUMN\_SERIAL** serial number (Valued = 2)  
**NUM\_COLUMNS** Number of the field, in this case Value = 3.

### 4.2.3 Function Documentation

#### 4.2.3.1 void add\_item (GtkWidget \* *widget*, GdkEvent \* *event*, gpointer *data*)

Add a item to treeview.

#### Parameters

- widget* Component.  
*event* Action.  
*data* Data passed to function.

#### 4.2.3.2 void authorize\_device (DEV \* *dev*)

Add a device to a database.

This function essentially add a new entry in a database with the data of dev.

#### Parameters

- dev* A device.

#### 4.2.3.3 GtkWidget\* do\_list\_store ()

Function that creates the main window.

Creates all the component that is present in the main window and itself.

##### Returns

Main window

#### 4.2.3.4 void fatal\_error\_signal (int sig)

handler for exceptions.

This function is called when a exception is take place in the system. When arrive a signal for to close the program this keep the signal in a circule since all clean task is performed. When all is done it set the default handler for that signal and allow to close the program.

##### Parameters

*sig* A in that identify the arrived signal.

#### 4.2.3.5 gchar\* get\_new\_device\_id ()

Get a identify new for a row.

This is a auxiliary function which is called by add\_item. This show a confirmation message.

##### Returns

Identify's device.

#### 4.2.3.6 int main (int argc, char \*\* argv)

main function.

This is the begin of the program.

##### Parameters

*argc* Number of the arguments.

*argv* List of the arguments.

##### Returns

#### 4.2.3.7 DEV\* read\_devices\_list ()

Create a list of [DEV](#) \* Read the database normaly in /database.db and add to [DEV](#) \* list the device there are.

##### Returns

[DEV](#) \* pointer



**4.2.3.8 void remove\_item (GtkWidget \* *widget*, GdkEvent \* *event*, gpointer *data*)**

Delete rows from treeview.

Search all rows these are marked and delete it.

**Parameters**

*widget* The object the represent the component.

*event* is the action that was performed.

*data* other data that is passed to method.

**4.2.3.9 void update\_tool\_bar ()**

Modify status of toolbar.

Change the status of the component for add and delete row in the treeview.

# Index

/media/4ec1cddc-24fd-4aaa-a1b9-43731b032863/home/editablez/workspace/usb/src/common.h, 7  
\_\_request, 5  
add\_item  
    usb-gui.h, 11  
authorize\_device  
    usb-gui.h, 11  
COLUMN\_ID  
    usb-gui.h, 11  
COLUMN\_MARKED  
    usb-gui.h, 11  
COLUMN\_SERIAL  
    usb-gui.h, 11  
common.h  
    get\_date, 8  
    getmsg, 9  
    getuser, 9  
    register\_log, 9  
DEV, 6  
do\_list\_store  
    usb-gui.h, 11  
fatal\_error\_signal  
    usb-gui.h, 12  
get\_date  
    common.h, 8  
get\_new\_device\_id  
    usb-gui.h, 12  
getmsg  
    common.h, 9  
getuser  
    common.h, 9  
main  
    usb-gui.h, 12  
NUM\_COLUMNS  
    usb-gui.h, 11  
read\_devices\_list  
    usb-gui.h, 12  
register\_log  
    common.h, 9  
remove\_item  
    usb-gui.h, 12  
update\_tool\_bar  
    usb-gui.h, 13  
usb-gui.h, 10  
    add\_item, 11  
    authorize\_device, 11  
    COLUMN\_ID, 11  
    COLUMN\_MARKED, 11  
    COLUMN\_SERIAL, 11  
    do\_list\_store, 11  
    fatal\_error\_signal, 12  
    get\_new\_device\_id, 12  
    main, 12  
    NUM\_COLUMNS, 11  
    read\_devices\_list, 12  
    remove\_item, 12  
    update\_tool\_bar, 13