gtkterm

Generated by Doxygen 1.9.4

Chapter 1

GTKTerm: A GTK+ Serial Port Terminal

GTKTerm is a simple, graphical serial port terminal emulator for Linux and possibly other POSIX-compliant operating systems. It can be used to communicate with all kinds of devices with a serial interface, such as embedded computers, microcontrollers, modems, GPS receivers, CNC machines and more.

1.1 Usage

1.1.1 Keyboard Shortcuts

As GTKTerm is often used like a terminal emulator, the shortcut keys are assigned to `<ctrl><shift>`, rather than just `<ctrl>`. This allows the user to send keystrokes of the form `<ctrl>X` and not have GTKTerm intercept them.

Key Combination	Effect
`<ctrl><shift>L`	Clear screen
`<ctrl><shift>R`	Send file
`<ctrl><shift>Q`	Quit
`<ctrl><shift>S`	Configure port
`<ctrl><shift>V`	Paste
`<ctrl><shift>C`	Copy
`<ctrl><shift>F`	Find
`<ctrl><shift>K`	Clear Scrollback
`<ctrl><shift>A`	Select All
`<ctrl><shift>B`	Send Break
`<ctrl>B`	Send break
F5	Open Port
F6	Close Port
F7	Toggle DTR
F8	Toggle RTS

1.1.2 Command Line Options

See `man gtkterm` or `gtkterm --help` for more information on available command line interface options.

1.1.3 Notes on RS485:

The RS485 flow control is a software user-space emulation and therefore may not work for all configurations (won't respond quickly enough). If this is the case for your setup, you will need to either use a dedicated RS232 to RS485 converter, or look for a kernel level driver. This is an inherent limitation to user space programs.

1.1.4 Scriptability with Signals

Some microcontrollers and other embedded devices are flashed using the same serial interface that is also used for outputting debug information. To facilitate rapid development on these platforms, GTKTerm supports the following UNIX signals:

Signal	Action	Usage Example
`SIGUSR1`	Open Port	`killall -USR1 gtkterm`
`SIGUSR2`	Close Port	`killall -USR2 gtkterm`

You may find it useful to send these signals in your own firmware flashing scripts.

1.2 Installation

GTKTerm has a few dependencies-

- Gtk+4.0 (version 4.6 or higher)
- vte-gtk4 (version 0.68 or higher)
- intltool (version 0.40.0 or higher)
- libgudev (version 229 or higher)

Once these dependencies are installed, most people should simply run:

```
meson build
ninja -C build
```

To install GTKTerm system-wide, run:

```
ninja -C build install
gtk-update-icon-cache
```

If you wish to install GTKTerm someplace other than the default directory, e.g. in `/usr`, use:

```
meson build -Dprefix=/usr
```

Then build and install as usual.

1.3 Uninstallation

To uninstall GTKTerm, run:

```
ninja -C build uninstall
```

If you already deleted the `build` directory, just compile and install GTKTerm again as explained in the previous section with the same target location prefix (`-Dprefix`) and perform the uninstall step afterwards.

1.4 License

Original Code by: Julien Schmitt

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

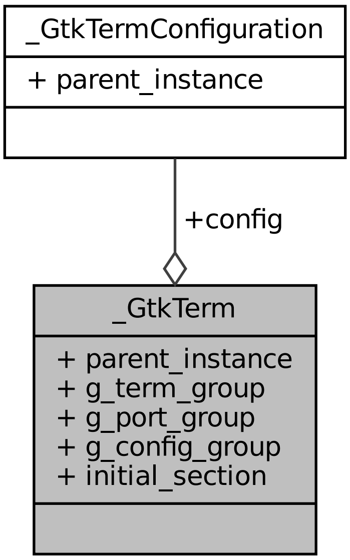
Class Documentation

4.1 _GtkTerm Struct Reference

The main GtkTerm application class.

```
#include <gtkterm.h>
```

Collaboration diagram for _GtkTerm:



Public Attributes

- GtkApplication parent_instance
- GOptionGroup ∗ g_term_group
- GOptionGroup ∗ g_port_group
- GOptionGroup ∗ g_config_group
- GtkTermConfiguration ∗ config
- char ∗ initial_section

  The Key file with the configurations.

4.1.1 Detailed Description

The main GtkTerm application class.

All application specific variables are defined here.

4.1.2 Member Data Documentation

4.1.2.1 config

```
GtkTermConfiguration* _GtkTerm::config
```

4.1.2.2 g_config_group

```
GOptionGroup* _GtkTerm::g_config_group
```

Referenced by gtkterm_add_cmdline_options().

4.1.2.3 g_port_group

```
GOptionGroup* _GtkTerm::g_port_group
```

Referenced by gtkterm_add_cmdline_options().

4.1.2.4 g_term_group

```
GOptionGroup* _GtkTerm::g_term_group
```

Referenced by gtkterm_add_cmdline_options().

4.1.2.5 initial_section

```
char* _GtkTerm::initial_section
```

The Key file with the configurations.

4.1.2.6 parent_instance

```
GtkApplication _GtkTerm::parent_instance
```

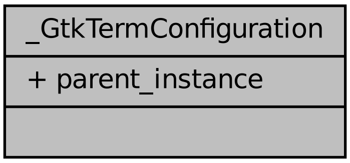
The documentation for this struct was generated from the following file:

- gtkterm.h

4.2 _GtkTermConfiguration Struct Reference

```
#include <resource_file.h>
```

Collaboration diagram for _GtkTermConfiguration:



Public Attributes

- GObject parent_instance

4.2.1 Member Data Documentation

4.2.1.1 parent_instance

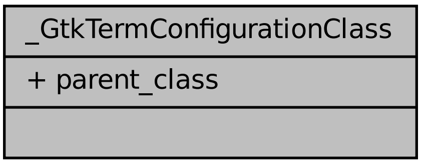
```
GObject _GtkTermConfiguration::parent_instance
```

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

- resource_file.h

4.3 _GtkTermConfigurationClass Struct Reference

Collaboration diagram for _GtkTermConfigurationClass:



Public Attributes

- GObjectClass parent_class

4.3.1 Member Data Documentation

4.3.1.1 parent_class

```
GObjectClass _GtkTermConfigurationClass::parent_class
```

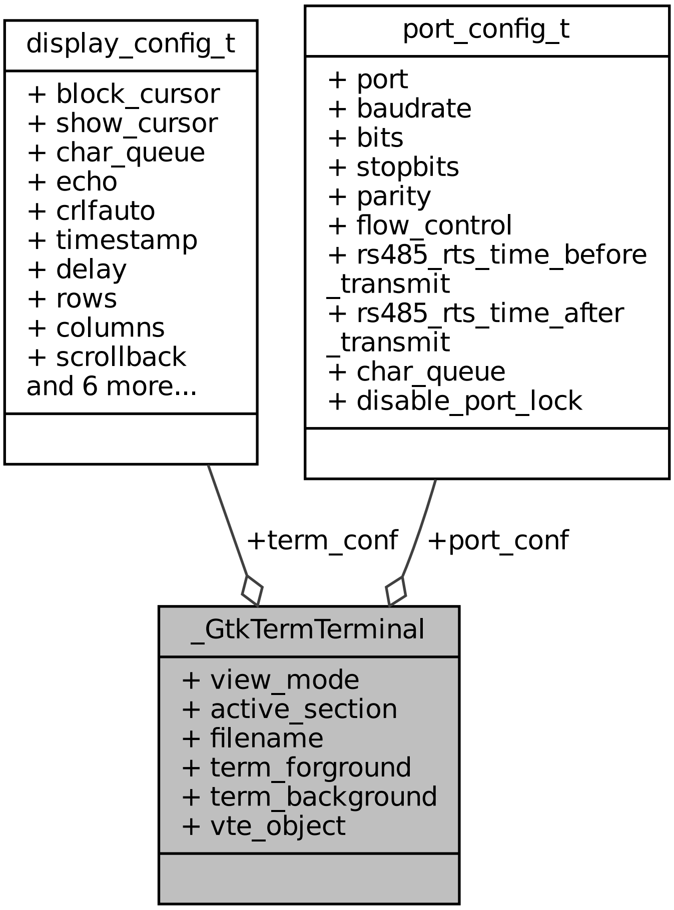
The documentation for this struct was generated from the following file:

- resource_file.c

4.4 _GtkTermTerminal Struct Reference

```
#include <terminal.h>
```

Collaboration diagram for _GtkTermTerminal:



Public Attributes

- uint8_t view_mode
- display_config_t ∗ term_conf

  ASCII or HEX view mode.
- port_config_t ∗ port_conf

  The configuration loaded from the keyfile.
- char ∗ active_section

  Port configuration used in this terminal.
- char ∗ filename

  Active section in this window from config file.
- GdkRGBA ∗ term_forground

  File to send.
- GdkRGBA ∗ term_background

  Foreground (text) color of this terminal.
- VteTerminal vte_object

  Background color.

4.4.1 Member Data Documentation

4.4.1.1 active_section

`char* _GtkTermTerminal::active_section`

Port configuration used in this terminal.

4.4.1.2 filename

`char* _GtkTermTerminal::filename`

Active section in this window from config file.

4.4.1.3 port_conf

`port_config_t* _GtkTermTerminal::port_conf`

The configuration loaded from the keyfile.

4.4.1.4 term_background

`GdkRGBA* _GtkTermTerminal::term_background`

Foreground (text) color of this terminal.

4.4.1.5 term_conf

`display_config_t* _GtkTermTerminal::term_conf`

ASCII or HEX view mode.

4.4.1.6 term_forground

```
GdkRGBA* _GtkTermTerminal::term_forground
```

File to send.

4.4.1.7 view_mode

```
uint8_t _GtkTermTerminal::view_mode
```

4.4.1.8 vte_object

```
VteTerminal _GtkTermTerminal::vte_object
```

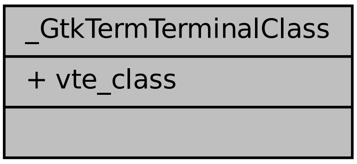
Background color.

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

- terminal.h

4.5 _GtkTermTerminalClass Struct Reference

Collaboration diagram for _GtkTermTerminalClass:



Public Attributes

- VteTerminalClass vte_class

4.5.1 Member Data Documentation

4.5.1.1 vte_class

```
VteTerminalClass _GtkTermTerminalClass::vte_class
```

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

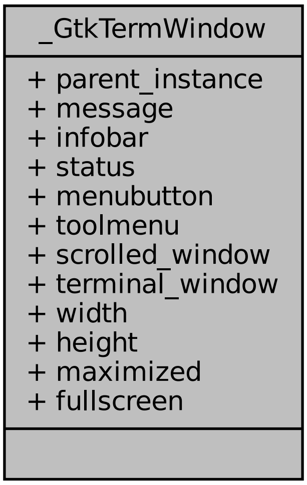
- terminal.c

4.6 _GtkTermWindow Struct Reference

The main GtkTermWindow class.

```
#include <gtkterm.h>
```

Collaboration diagram for _GtkTermWindow:



Public Attributes

- GtkApplicationWindow parent_instance
- GtkWidget ∗ message
- GtkWidget ∗ infobar

    Message for the infobar.
- GtkWidget ∗ status

    Infobar.
- GtkWidget ∗ menubutton

    Statusbar.
- GMenuModel ∗ toolmenu

    Toolbar.

- GtkScrolledWindow * scrolled_window

  Menu.
- GtkTermTerminal * terminal_window

  Make the terminal window scrolled.
- int width

  The terminal window.
- int height
- bool maximized
- bool fullscreen

4.6.1 Detailed Description

The main GtkTermWindow class.

MainWindow specific variables here.

4.6.2 Member Data Documentation

4.6.2.1 fullscreen

```
bool _GtkTermWindow::fullscreen
```

4.6.2.2 height

```
int _GtkTermWindow::height
```

4.6.2.3 infobar

```
GtkWidget* _GtkTermWindow::infobar
```

Message for the infobar.

4.6.2.4 maximized

```
bool _GtkTermWindow::maximized
```

4.6.2.5 menubutton

```
GtkWidget* _GtkTermWindow::menubutton
```

Statusbar.

4.6.2.6 message

```
GtkWidget* _GtkTermWindow::message
```

4.6.2.7 parent_instance

```
GtkApplicationWindow _GtkTermWindow::parent_instance
```

4.6.2.8 scrolled_window

```
GtkScrolledWindow* _GtkTermWindow::scrolled_window
```

Menu.

4.6.2.9 status

```
GtkWidget* _GtkTermWindow::status
```

Infobar.

4.6.2.10 terminal_window

```
GtkTermTerminal* _GtkTermWindow::terminal_window
```

Make the terminal window scrolled.

4.6.2.11 toolmenu

```
GMenuModel* _GtkTermWindow::toolmenu
```

Toolbar.

4.6.2.12 width

```
int _GtkTermWindow::width
```

The terminal window.

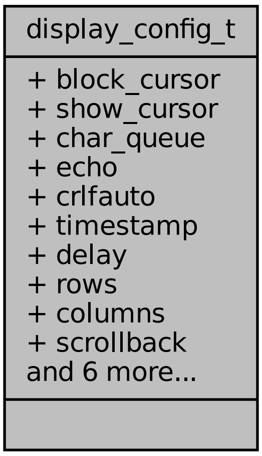
The documentation for this struct was generated from the following file:

- gtkterm.h

4.7 display_config_t Struct Reference

```
#include <term_config.h>
```

Collaboration diagram for display_config_t:



Public Attributes

- bool block_cursor
- bool show_cursor
- char char_queue
- bool echo
- bool crlfauto
- bool timestamp
- int delay
- int rows
- int columns
- int scrollback
- bool visual_bell
- GdkRGBA foreground_color
- GdkRGBA background_color
- PangoFontDescription ∗ font
- char ∗ active_section
- char ∗ default_filename

4.7.1 Member Data Documentation

4.7.1.1 active_section

```
char* display_config_t::active_section
```

4.7.1.2 background_color

```
GdkRGBA display_config_t::background_color
```

4.7.1.3 block_cursor

```
bool display_config_t::block_cursor
```

4.7.1.4 char_queue

```
char display_config_t::char_queue
```

4.7.1.5 columns

```
int display_config_t::columns
```

4.7.1.6 crlfauto

```
bool display_config_t::crlfauto
```

4.7.1.7 default_filename

```
char* display_config_t::default_filename
```

4.7.1.8 delay

```
int display_config_t::delay
```

4.7.1.9 echo

```
bool display_config_t::echo
```

4.7.1.10 font

```
PangoFontDescription* display_config_t::font
```

4.7.1.11 foreground_color

```
GdkRGBA display_config_t::foreground_color
```

4.7.1.12 rows

```
int display_config_t::rows
```

4.7.1.13 scrollback

```
int display_config_t::scrollback
```

4.7.1.14 show_cursor

```
bool display_config_t::show_cursor
```

4.7.1.15 timestamp

```
bool display_config_t::timestamp
```

4.7.1.16 visual_bell

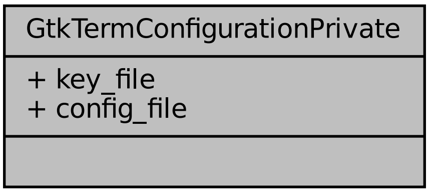
```
bool display_config_t::visual_bell
```

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

- term_config.h

4.8 GtkTermConfigurationPrivate Struct Reference

Collaboration diagram for GtkTermConfigurationPrivate:



Public Attributes

- GKeyFile ∗ key_file
- GFile ∗ config_file

  The memory loaded keyfile.

4.8.1 Member Data Documentation

4.8.1.1 config_file

```
GFile∗ GtkTermConfigurationPrivate::config_file
```

The memory loaded keyfile.

4.8.1.2 key_file

```
GKeyFile∗ GtkTermConfigurationPrivate::key_file
```

Referenced by gtkterm_configuration_default_configuration(), gtkterm_configuration_validate(), and on_set_config_options().

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

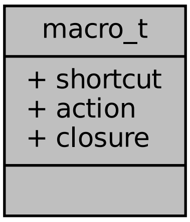
- resource_file.c

4.9 macro_t Struct Reference

TODO: Migrate to GObject.

```
#include <macros.h>
```

Collaboration diagram for macro_t:



Public Attributes

- char ∗ shortcut
- char ∗ action

  Shortcut of the macro.
- GClosure ∗ closure

  Command to perform.

4.9.1 Detailed Description

TODO: Migrate to GObject.

Define macro structure type

4.9.2 Member Data Documentation

4.9.2.1 action

```
char* macro_t::action
```

Shortcut of the macro.

Referenced by convert_macros_to_string(), and convert_string_to_macros().

4.9.2.2 closure

```
GClosure* macro_t::closure
```

Command to perform.

4.9.2.3 shortcut

```
char* macro_t::shortcut
```

Referenced by convert_macros_to_string(), and convert_string_to_macros().

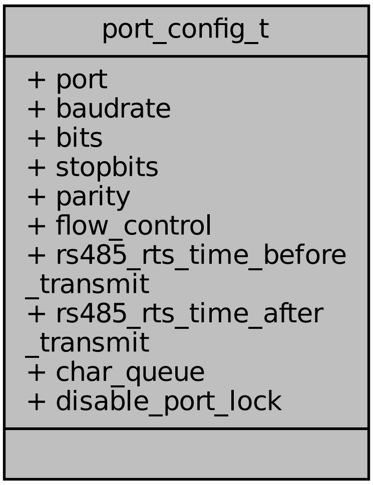
The documentation for this struct was generated from the following file:

- macros.h

4.10 port_config_t Struct Reference

```
#include <serial.h>
```

Collaboration diagram for port_config_t:



Public Attributes

- char port [256]
- long int baudrate
- int bits
- int stopbits
- int parity
- int flow_control
- int rs485_rts_time_before_transmit
- int rs485_rts_time_after_transmit
- char char_queue
- bool disable_port_lock

4.10.1 Member Data Documentation

4.10.1.1 baudrate

```
long int port_config_t::baudrate
```

Referenced by get_port_string().

4.10.1.2 bits

```
int port_config_t::bits
```

Referenced by get_port_string().

4.10.1.3 char_queue

```
char port_config_t::char_queue
```

4.10.1.4 disable_port_lock

```
bool port_config_t::disable_port_lock
```

4.10.1.5 flow_control

```
int port_config_t::flow_control
```

4.10.1.6 parity

```
int port_config_t::parity
```

Referenced by get_port_string().

4.10.1.7 port

```
char port_config_t::port[256]
```

Referenced by get_port_string().

4.10.1.8 rs485_rts_time_after_transmit

```
int port_config_t::rs485_rts_time_after_transmit
```

4.10.1.9 rs485_rts_time_before_transmit

```
int port_config_t::rs485_rts_time_before_transmit
```

4.10.1.10 stopbits

```
int port_config_t::stopbits
```

Referenced by get_port_string().

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

- serial.h

Chapter 5

File Documentation

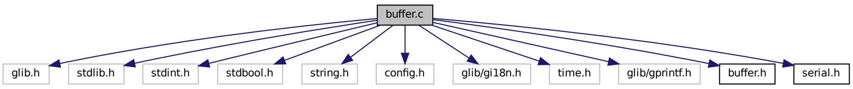
5.1 README.md File Reference

5.2 README.source File Reference

5.3 buffer.c File Reference

```
#include <glib.h>
#include <stdlib.h>
#include <stdint.h>
#include <stdbool.h>
#include <string.h>
#include <config.h>
#include <glib/gi18n.h>
#include <time.h>
#include <glib/gprintf.h>
#include "buffer.h"
#include "serial.h"
```

Include dependency graph for buffer.c:



Macros

- #define TIMESTAMP_SIZE 50

Functions

- void create_buffer (void)
- void delete_buffer (void)
- unsigned int insert_timestamp (char ∗buffer)
- void put_chars (const char ∗chars, unsigned int size, bool crlf_auto)
- void write_buffer (void)
- void write_buffer_with_func (void(∗func)(const char ∗, unsigned int))
- void clear_buffer (void)
- void set_clear_func (void(∗func)(void))
- void unset_clear_func (void(∗func)(void))
- void set_display_func (void(∗func)(const char ∗, unsigned int))
- void unset_display_func (void(∗func)(const char ∗, unsigned int))

Variables

- bool timestamp_on
- char overlapped
- unsigned int virt_col_pos
- void(∗ write_func)(const char ∗, unsigned int) = NULL
- void(∗ clear_func)(void) = NULL

5.3.1 Macro Definition Documentation

5.3.1.1 TIMESTAMP_SIZE

```
#define TIMESTAMP_SIZE 50
```

5.3.2 Function Documentation

5.3.2.1 clear_buffer()

```
void clear_buffer (
             void   )
```

References clear_func.

5.3.2.2 create_buffer()

```
void create_buffer (
             void   )
```

5.3.2.3 delete_buffer()

```
void delete_buffer (
             void  )
```

5.3.2.4 insert_timestamp()

```
unsigned int insert_timestamp (
             char * buffer )
```

5.3.2.5 put_chars()

```
void put_chars (
             const char * chars,
             unsigned int size,
             bool crlf_auto )
```

References RECEIVE_BUFFER, timestamp_on, and TIMESTAMP_SIZE.

5.3.2.6 set_clear_func()

```
void set_clear_func (
             void(*)(void) func )
```

References clear_func.

5.3.2.7 set_display_func()

```
void set_display_func (
             void(*)(const char *, unsigned int) func )
```

References write_func.

5.3.2.8 unset_clear_func()

```
void unset_clear_func (
             void(*)(void) func )
```

References clear_func.

5.3.2.9 unset_display_func()

```
void unset_display_func (
             void(*)(const char *, unsigned int) func )
```

References write_func.

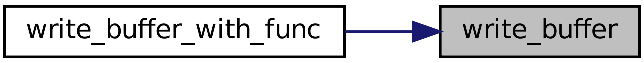
5.3.2.10 write_buffer()

```
void write_buffer (
             void )
```

References overlapped, and write_func.

Referenced by write_buffer_with_func().

Here is the caller graph for this function:

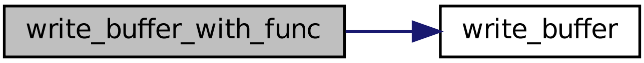


5.3.2.11 write_buffer_with_func()

```
void write_buffer_with_func (
             void(*)(const char *, unsigned int) func )
```

References write_buffer(), and write_func.

Here is the call graph for this function:



5.3.3 Variable Documentation

5.3.3.1 clear_func

```
void(* clear_func) (void) (
             void  ) = NULL
```

Referenced by clear_buffer(), set_clear_func(), and unset_clear_func().

5.3.3.2 overlapped

```
char overlapped
```

Referenced by write_buffer().

5.3.3.3 timestamp_on

```
bool timestamp_on  [extern]
```

Referenced by put_chars().

5.3.3.4 virt_col_pos

```
unsigned int virt_col_pos  [extern]
```

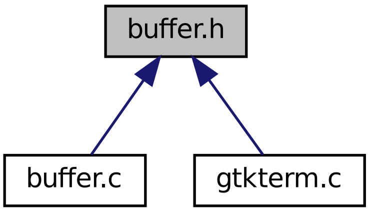
5.3.3.5 write_func

```
void(* write_func) (const char *, unsigned int) (
             const char * ,
             unsigned int  ) = NULL
```

Referenced by set_display_func(), unset_display_func(), write_buffer(), and write_buffer_with_func().

5.4 buffer.h File Reference

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



Macros

- #define BUFFER_SIZE (128 ∗ 1024)

Functions

- void create_buffer (void)
- void delete_buffer (void)
- void put_chars (const char ∗, unsigned int, bool)
- void clear_buffer (void)
- void write_buffer (void)
- void set_display_func (void(∗func)(const char ∗, uint32_t))
- void unset_display_func (void(∗func)(const char ∗, uint32_t))
- void set_clear_func (void(∗func)(void))
- void unset_clear_func (void(∗func)(void))
- void write_buffer_with_func (void(∗func)(const char ∗, uint32_t))

5.4.1 Macro Definition Documentation

5.4.1.1 BUFFER_SIZE

```
#define BUFFER_SIZE (128 * 1024)
```

5.4.2 Function Documentation

5.4.2.1 clear_buffer()

```
void clear_buffer (
             void  )
```

References clear_func.

5.4.2.2 create_buffer()

```
void create_buffer (
             void  )
```

5.4.2.3 delete_buffer()

```
void delete_buffer (
             void  )
```

5.4.2.4 put_chars()

```
void put_chars (
             const char * chars,
             unsigned int size,
             bool crlf_auto )
```

References RECEIVE_BUFFER, timestamp_on, and TIMESTAMP_SIZE.

5.4.2.5 set_clear_func()

```
void set_clear_func (
             void(*)(void)  func )
```

References clear_func.

5.4.2.6 set_display_func()

```
void set_display_func (
             void(*)(const char *, uint32_t)  func )
```

5.4.2.7 unset_clear_func()

```
void unset_clear_func (
            void(*)(void) func )
```

References clear_func.

5.4.2.8 unset_display_func()

```
void unset_display_func (
            void(*)(const char *, uint32_t) func )
```

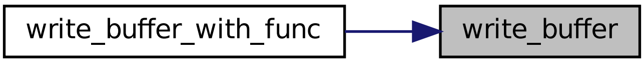
5.4.2.9 write_buffer()

```
void write_buffer (
            void )
```

References overlapped, and write_func.

Referenced by write_buffer_with_func().

Here is the caller graph for this function:



5.4.2.10 write_buffer_with_func()

```
void write_buffer_with_func (
            void(*)(const char *, uint32_t) func )
```

5.5 buffer.h

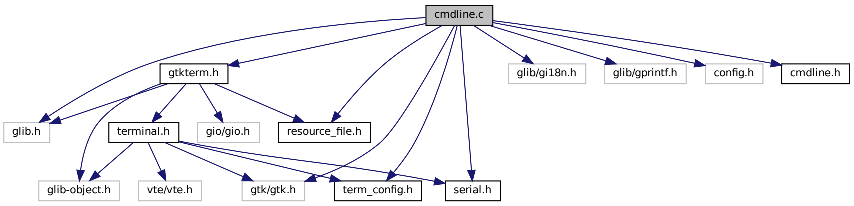
Go to the documentation of this file.

```
/***********************************************************************/
/* buffer.h                                                            */
/* --------                                                            */
/*           GTKTerm Software                                          */
/*                      (c) Julien Schmitt                             */
/*                                                                     */
/* ------------------------------------------------------------------- */
/*                                                                     */
/*   Purpose                                                           */
/*      Management of a local buffer of data received                  */
/*      - Header file -                                                */
/*                                                                     */
/*   ChangeLog                                                         */
/*      - 0.99.7 :  removed auto crlf stuff - (use macros instead)     */
/*      - 0.98.4 :  file creation by Julien                             */
/*                                                                     */
/***********************************************************************/

#ifndef BUFFER_H_
#define BUFFER_H_

#define BUFFER_SIZE (128 * 1024)

void create_buffer(void);
void delete_buffer(void);
void put_chars(const char *, unsigned int, bool);
void clear_buffer(void);
void write_buffer(void);
void set_display_func(void (*func)(const char *, uint32_t));
void unset_display_func(void (*func)(const char *, uint32_t));
void set_clear_func(void (*func)(void));
void unset_clear_func(void (*func)(void));
void write_buffer_with_func(void (*func)(const char *, uint32_t));

#endif
```

5.6 cmdline.c File Reference

```
#include <glib.h>
#include <glib/gi18n.h>
#include <gtk/gtk.h>
#include <glib/gprintf.h>
#include <config.h>
#include "gtkterm.h"
#include "resource_file.h"
#include "cmdline.h"
#include "term_config.h"
#include "serial.h"
```

Include dependency graph for cmdline.c:



Macros

- #define BUFFER_LENGTH 256
- #define MAX_SECTION_LENGTH 32

Functions

- void gtkterm_add_cmdline_options (GtkTerm ∗app)

5.6.1 Macro Definition Documentation

5.6.1.1 BUFFER_LENGTH

```
#define BUFFER_LENGTH 256
```

5.6.1.2 MAX_SECTION_LENGTH

```
#define MAX_SECTION_LENGTH 32
```

5.6.2 Function Documentation

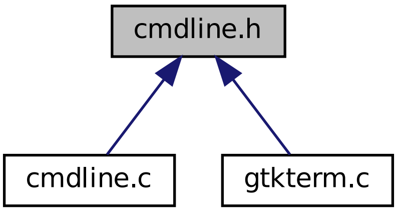
5.6.2.1 gtkterm_add_cmdline_options()

```
void gtkterm_add_cmdline_options (
            GtkTerm ∗ app )
```

References BUFFER_LENGTH, _GtkTerm::g_config_group, _GtkTerm::g_port_group, and _GtkTerm::g_term_group.

5.7 cmdline.h File Reference

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



Functions

- void gtkterm_add_cmdline_options (GtkTerm ∗app)

Variables

- GOptionGroup ∗ g_term_group

5.7.1 Function Documentation

5.7.1.1 gtkterm_add_cmdline_options()

```
void gtkterm_add_cmdline_options (
            GtkTerm ∗ app )
```

References BUFFER_LENGTH, _GtkTerm::g_config_group, _GtkTerm::g_port_group, and _GtkTerm::g_term_group.

5.7.2 Variable Documentation

5.7.2.1 g_term_group

```
GOptionGroup∗ g_term_group  [extern]
```

5.8 cmdline.h

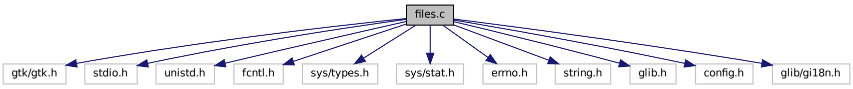
Go to the documentation of this file.

```
/***********************************************************************/
/* cmdline.h                                                           */
/* ---------                                                           */
/*           GTKTerm Software                                          */
/*                      (c) Julien Schmitt                             */
/*                                                                     */
/* ------------------------------------------------------------------- */
/*                                                                     */
/*   Purpose                                                           */
/*      Reads the command line                                         */
/*      - Header file -                                                */
/*                                                                     */
/*   ChangeLog                                                         */
/*      - 2.0 :  migrated to GTK4                                       */
/*      - 0.98 :  file creation by Julien                               */
/*                                                                     */
/***********************************************************************/extern GOptionGroup
    *g_term_group;

#ifndef CMDLINE_H
#define CMDLINE_H

void gtkterm_add_cmdline_options (GtkTerm *app);

#endif // CMDLINE_H
```

5.9 files.c File Reference

```
#include <gtk/gtk.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <errno.h>
#include <string.h>
#include <glib.h>
#include <config.h>
#include <glib/gi18n.h>
```

Include dependency graph for files.c:



Variables

- char ∗ default_filename = NULL

5.9.1 Variable Documentation

5.9.1.1 default_filename

```
char* default_filename = NULL
```

5.10 files.h File Reference

Functions

- void send_raw_file (GAction ∗action, gpointer data)
- void save_raw_file (GAction ∗action, gpointer data)
- void add_input (void)

Variables

- gboolean waiting_for_char
- char ∗ default_filename

5.10.1 Function Documentation

5.10.1.1 add_input()

```
void add_input (
            void  )
```

5.10.1.2 save_raw_file()

```
void save_raw_file (
            GAction * action,
            gpointer data )
```

5.10.1.3 send_raw_file()

```
void send_raw_file (
            GAction * action,
            gpointer data )
```

5.10.2 Variable Documentation

5.10.2.1 default_filename

```
char* default_filename  [extern]
```

5.10.2.2 waiting_for_char

```
gboolean waiting_for_char  [extern]
```

5.11 files.h

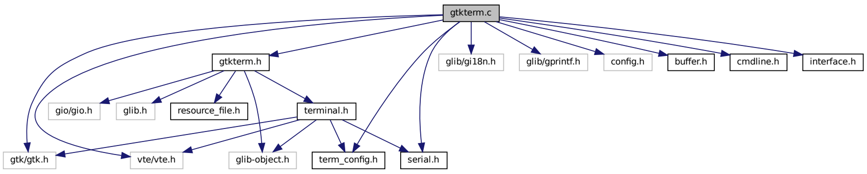
Go to the documentation of this file.

```
/***********************************************************************/
/* files.h                                                           */
/* ---------                                                           */
/*           GTKTerm Software                                          */
/*                      (c) Julien Schmitt                             */
/*                                                                     */
/* ------------------------------------------------------------------- */
/*                                                                     */
/*   Purpose                                                           */
/*      Raw / text file transfer management                            */
/*      - Header file -                                                */
/*                                                                     */
/***********************************************************************/

#ifndef FILES_H_
#define FILES_H_

void send_raw_file (GAction *action, gpointer data);
void save_raw_file (GAction *action, gpointer data);
void add_input(void);

extern gboolean waiting_for_char;
extern char *default_filename;

#endif
```

5.12 gtkterm.c File Reference

```
#include <gtk/gtk.h>
#include <vte/vte.h>
#include <glib/gi18n.h>
#include <glib/gprintf.h>
#include "config.h"
#include "gtkterm.h"
#include "buffer.h"
#include "cmdline.h"
#include "interface.h"
#include "term_config.h"
#include "serial.h"
```

Include dependency graph for gtkterm.c:



Macros

- #define NR_OF_SIGNALS 4
- #define GTKTERM_TYPE gtkterm_get_type ()
- #define GTKTERM_WINDOW_TYPE gtkterm_window_get_type ()

Functions

- void set_window_title (GtkTermWindow ∗)
- int main (int argc, char ∗argv[ ])

Variables

- unsigned int gtkterm_signals [NR_OF_SIGNALS]

5.12.1 Macro Definition Documentation

5.12.1.1 GTKTERM_TYPE

```
#define GTKTERM_TYPE gtkterm_get_type ()
```

5.12.1.2 GTKTERM_WINDOW_TYPE

```
#define GTKTERM_WINDOW_TYPE gtkterm_window_get_type ()
```

5.12.1.3 NR_OF_SIGNALS

```
#define NR_OF_SIGNALS 4
```

5.12.2 Function Documentation

5.12.2.1 main()

```
int main (
             int argc,
             char * argv[ ] )
```

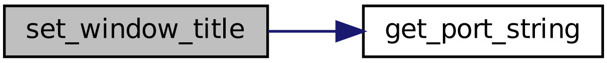
References GTKTERM_TYPE.

5.12.2.2 set_window_title()

```
void set_window_title (
            GtkTermWindow * window )
```

References get_port_string().

Here is the call graph for this function:



5.12.3 Variable Documentation

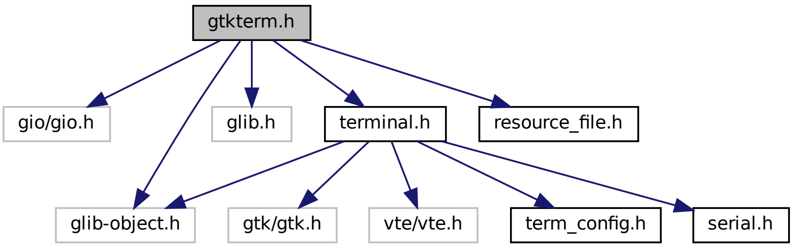
5.12.3.1 gtkterm_signals

```
unsigned int gtkterm_signals[NR_OF_SIGNALS]
```

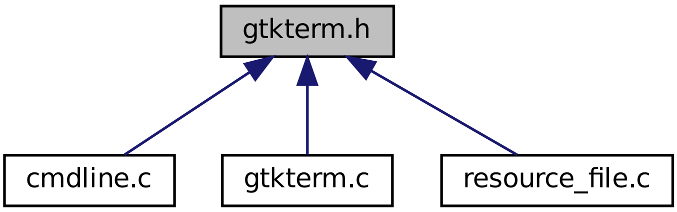
5.13 gtkterm.h File Reference

```
#include <gio/gio.h>
#include <glib-object.h>
#include <glib.h>
#include "terminal.h"
#include "resource_file.h"
```

Include dependency graph for gtkterm.h:



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



Classes

- struct _GtkTerm

  The main GtkTerm application class.
- struct _GtkTermWindow

  The main GtkTermWindow class.

Typedefs

- typedef struct _GtkTerm GtkTerm
- typedef struct _GtkTermWindow GtkTermWindow

Enumerations

- enum { SIGNAL_LOAD_CONFIG , SIGNAL_SAVE_CONFIG , SIGNAL_REMOVE_SECTION , SIGNAL_PRINT_SECTION }

Variables

- unsigned int gtkterm_signals [ ]

5.13.1 Typedef Documentation

5.13.1.1 GtkTerm

```
typedef struct _GtkTerm GtkTerm
```

5.13.1.2 GtkTermWindow

```
typedef struct _GtkTermWindow GtkTermWindow
```

5.13.2 Enumeration Type Documentation

5.13.2.1 anonymous enum

```
anonymous enum
```

Enumerator

SIGNAL_LOAD_CONFIG	
SIGNAL_SAVE_CONFIG	
SIGNAL_REMOVE_SECTION	
SIGNAL_PRINT_SECTION	

5.13.3 Variable Documentation

5.13.3.1 gtkterm_signals

```
unsigned int gtkterm_signals[ ] [extern]
```

5.14 gtkterm.h

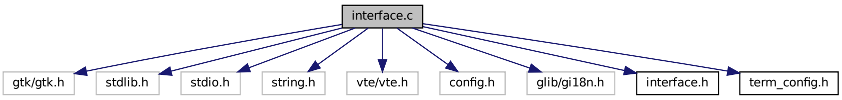
Go to the documentation of this file.

```

#ifndef GTKTERM_H
#define GTKTERM_H

#include <gio/gio.h>
#include <glib-object.h>
#include <glib.h>

#include "terminal.h"
#include "resource_file.h"

enum {
    SIGNAL_LOAD_CONFIG,
    SIGNAL_SAVE_CONFIG,
    SIGNAL_REMOVE_SECTION,
    SIGNAL_PRINT_SECTION
};

extern unsigned int gtkterm_signals[];

G_BEGIN_DECLS

//!  @brief The main GtkTerm application class.
//!  All application specific variables are defined here.
struct _GtkTerm {

  GtkApplication parent_instance;

  GOptionGroup *g_term_group;
  GOptionGroup *g_port_group;
  GOptionGroup *g_config_group;

  GtkTermConfiguration *config;          //!  The Key file with the configurations
  char *initial_section;                 //!  The initial section provided from the cli.
//!  Terminals have their own section pointer

};

typedef struct _GtkTerm GtkTerm;
G_DECLARE_FINAL_TYPE (GtkTerm, gtkterm, GTKTERM, APP, GtkApplication)

//!  @brief The main GtkTermWindow class.
//!  MainWindow specific variables here.
struct _GtkTermWindow {
  GtkApplicationWindow parent_instance;

  GtkWidget *message;                    //!  Message for the infobar
  GtkWidget *infobar;                    //!  Infobar
  GtkWidget *status;                     //!  Statusbar
  GtkWidget *menubutton;                 //!  Toolbar
  GMenuModel *toolmenu;                  //!  Menu
  GtkScrolledWindow *scrolled_window;    //!  Make the terminal window scrolled
  GtkTermTerminal *terminal_window;      //!  The terminal window

  int width;
  int height;
  bool maximized;
  bool fullscreen;

} ;

typedef struct _GtkTermWindow GtkTermWindow ;
G_DECLARE_FINAL_TYPE (GtkTermWindow, gtkterm_window, GTKTERM, WINDOW, GtkApplicationWindow)

G_END_DECLS

#endif // GTKTERM_H
```

5.15 interface.c File Reference

```
#include <gtk/gtk.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <vte/vte.h>
#include <config.h>
#include <glib/gi18n.h>
#include "interface.h"
#include "term_config.h"
```

Include dependency graph for interface.c:



Functions

- void show_message (char *message, int type_msg)

Variables

- bool timestamp_on = 0
- struct configuration_port config
- int virt_col_pos = 0

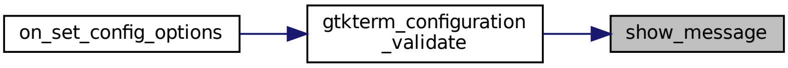
5.15.1 Function Documentation

5.15.1.1 show_message()

```
void show_message (
             char * message,
             int type_msg )
```

Referenced by gtkterm_configuration_validate().

Here is the caller graph for this function:



5.15.2 Variable Documentation

5.15.2.1 config

```
struct configuration_port config  [extern]
```

Referenced by on_set_config_options().

5.15.2.2 timestamp_on

```
bool timestamp_on = 0
```

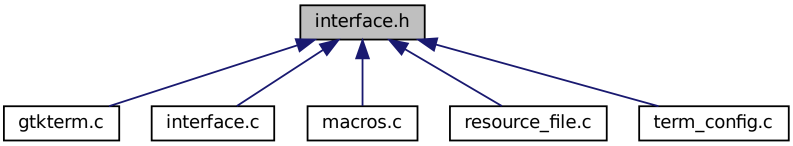
Referenced by put_chars().

5.15.2.3 virt_col_pos

```
int virt_col_pos = 0
```

5.16 interface.h File Reference

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



Macros

- #define MSG_WRN 0
- #define MSG_ERR 1
- #define ASCII_VIEW 0
- #define HEXADECIMAL_VIEW 1

Functions

- void show_message (char ∗, int)

Variables

- GtkWidget ∗ Text
- GtkWidget ∗ display

5.16.1 Macro Definition Documentation

5.16.1.1 ASCII_VIEW

```
#define ASCII_VIEW 0
```

5.16.1.2 HEXADECIMAL_VIEW

```
#define HEXADECIMAL_VIEW 1
```

5.16.1.3 MSG_ERR

```
#define MSG_ERR 1
```

5.16.1.4 MSG_WRN

```
#define MSG_WRN 0
```

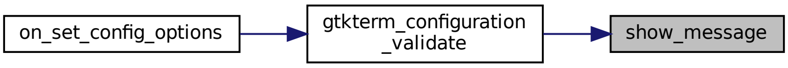
5.16.2 Function Documentation

5.16.2.1 show_message()

```
void show_message (
            char * message,
            int type_msg )
```

Referenced by gtkterm_configuration_validate().

Here is the caller graph for this function:



5.16.3 Variable Documentation

5.16.3.1 display

```
GtkWidget* display  [extern]
```

5.16.3.2 Text

```
GtkWidget* Text  [extern]
```

5.17 interface.h

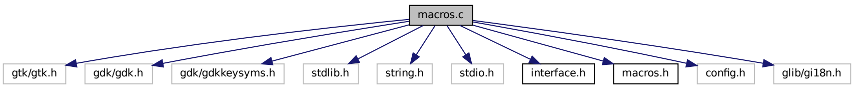
Go to the documentation of this file.

```
/***********************************************************************/
/* interface.h                                                         */
/* ---------                                                           */
/*           GTKTerm Software                                          */
/*                      (c) Julien Schmitt                             */
/*                                                                     */
/* ------------------------------------------------------------------- */
/*                                                                     */
/*   Purpose                                                           */
/*      Functions for the management of the GUI for the main window    */
/*      - Header file -                                                */
/*                                                                     */
/***********************************************************************/

#ifndef WIDGETS_H_
#define WIDGETS_H_

#define MSG_WRN          0
#define MSG_ERR          1

#define ASCII_VIEW       0
#define HEXADECIMAL_VIEW 1

extern GtkWidget *Text;
extern GtkWidget *display;          // Serial terminal (vte)

void show_message(char *, int);

#endif
```

5.18 macros.c File Reference

```
#include <gtk/gtk.h>
#include <gdk/gdk.h>
#include <gdk/gdkkeysyms.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include "interface.h"
#include "macros.h"
#include <config.h>
#include <glib/gi18n.h>
```

Include dependency graph for macros.c:



Enumerations

- enum { COLUMN_SHORTCUT , COLUMN_ACTION , NUM_COLUMNS }

  TODO: Migrate to GObject.

Functions

- int macro_count ()
- void convert_string_to_macros (char ∗∗string_list, int size)

  Convert the array of strings to macros.
- int convert_macros_to_string (char ∗∗string_list)

  Convert the in memory macros to an array of strings for storage in file.
- macro_t ∗ get_shortcuts (int ∗size)
- void remove_shortcuts (void)

Variables

- macro_t ∗ macros = NULL
- int nr_of_macros = 0

5.18.1 Enumeration Type Documentation

5.18.1.1 anonymous enum

```
anonymous enum
```

TODO: Migrate to GObject.

Enumerator

COLUMN_SHORTCUT	
COLUMN_ACTION	
NUM_COLUMNS	

5.18.2 Function Documentation

5.18.2.1 convert_macros_to_string()

```
int convert_macros_to_string (
            char ** string_list )
```

Convert the in memory macros to an array of strings for storage in file.

Must be NULL terminated

Number of strings is 2x the macros (shortcut and action)

References macro_t::action, macros, nr_of_macros, and macro_t::shortcut.

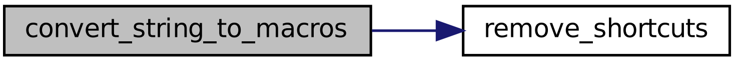
5.18.2.2 convert_string_to_macros()

```
void convert_string_to_macros (
             char ** string_list,
             int size )
```

Convert the array of strings to macros.

References macro_t::action, macros, nr_of_macros, remove_shortcuts(), and macro_t::shortcut.

Here is the call graph for this function:



5.18.2.3 get_shortcuts()

```
macro_t * get_shortcuts (
             int * size )
```

References macros.

5.18.2.4 macro_count()

```
int macro_count ( )
```

References nr_of_macros.

5.18.2.5 remove_shortcuts()

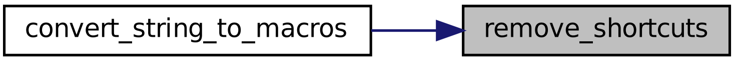
```
void remove_shortcuts (
             void  )
```

Clean up all macros

References macros.

Referenced by convert_string_to_macros().

Here is the caller graph for this function:



5.18.3 Variable Documentation

5.18.3.1 macros

```
macro_t* macros = NULL
```

Referenced by convert_macros_to_string(), convert_string_to_macros(), get_shortcuts(), and remove_shortcuts().

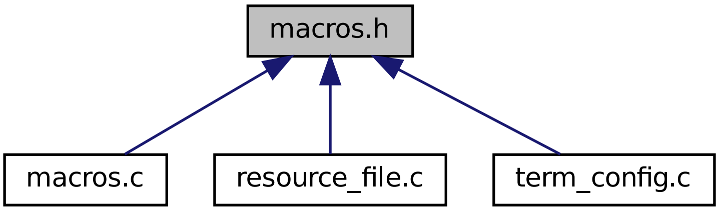
5.18.3.2 nr_of_macros

```
int nr_of_macros = 0
```

Referenced by convert_macros_to_string(), convert_string_to_macros(), and macro_count().

5.19 macros.h File Reference

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



Classes

- struct macro_t

  TODO: Migrate to GObject.

Functions

- void remove_shortcuts (void)
- void add_shortcuts (void)

  Remove shortcuts from accel_group and free memory.
- macro_t ∗ get_shortcuts (gint ∗)
- void convert_string_to_macros (char ∗∗, int)

  Convert the array of strings to macros.
- int convert_macros_to_string (char ∗∗)

  Convert the in memory macros to an array of strings for storage in file.
- int macro_count ()

Variables

- macro_t ∗ macros

5.19.1 Function Documentation

5.19.1.1 add_shortcuts()

```
void add_shortcuts (
             void   )
```

Remove shortcuts from accel_group and free memory.

5.19.1.2 convert_macros_to_string()

```
int convert_macros_to_string (
             char ** string_list )
```

Convert the in memory macros to an array of strings for storage in file.

Must be NULL terminated

Number of strings is 2x the macros (shortcut and action)

References macro_t::action, macros, nr_of_macros, and macro_t::shortcut.

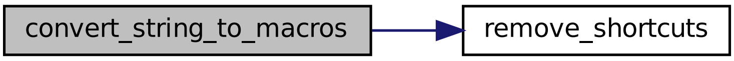
5.19.1.3 convert_string_to_macros()

```
void convert_string_to_macros (
             char ** string_list,
             int size )
```

Convert the array of strings to macros.

References macro_t::action, macros, nr_of_macros, remove_shortcuts(), and macro_t::shortcut.

Here is the call graph for this function:



5.19.1.4 get_shortcuts()

```
macro_t * get_shortcuts (
              gint *  )
```

5.19.1.5 macro_count()

```
int macro_count ( )
```

References nr_of_macros.

5.19.1.6 remove_shortcuts()

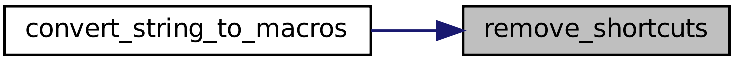
```
void remove_shortcuts (
              void  )
```

Clean up all macros

References macros.

Referenced by convert_string_to_macros().

Here is the caller graph for this function:



5.19.2 Variable Documentation

5.19.2.1 macros

```
macro_t* macros  [extern]
```

Referenced by convert_macros_to_string(), convert_string_to_macros(), get_shortcuts(), and remove_shortcuts().

5.20 macros.h

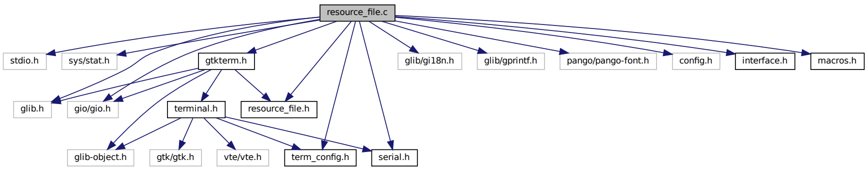
Go to the documentation of this file.

```
/***********************************************************************
 * macros.h
 * --------
 *           GTKTerm Software
 *                      (c) Julien Schmitt
 *
 * -------------------------------------------------------------------
 *
 * \brief Purpose
 *        Functions for the management of the macros
 *        - Header file -
 *
 ***********************************************************************/

#ifndef MACROS_H_
#define MACROS_H_

//!  TODO: Migrate to GObject

//!  Define macro structure type
typedef struct
{
    char *shortcut;     //!  Shortcut of the macro
    char *action;     //!  Command to perform
    GClosure *closure;   //!
}
macro_t;

//void config_macros(GtkAction *action, gpointer data);
void remove_shortcuts(void);                //!  Remove shortcuts from accel_group and free memory
void add_shortcuts(void);                  //!
macro_t *get_shortcuts(gint *);             //!

void convert_string_to_macros (char **, int);
int convert_macros_to_string (char **);

int macro_count ();

extern macro_t *macros;

#endif
```

5.21 resource_file.c File Reference

```
#include <stdio.h>
#include <sys/stat.h>
#include <glib.h>
#include <glib/gi18n.h>
#include <glib/gprintf.h>
#include <gio/gio.h>
#include <pango/pango-font.h>
#include <config.h>
#include "gtkterm.h"
#include "serial.h"
#include "term_config.h"
#include "resource_file.h"
#include "interface.h"
#include "macros.h"
```

Include dependency graph for resource_file.c:



Classes

- struct GtkTermConfigurationPrivate
- struct _GtkTermConfigurationClass

Macros

- #define CONFIGURATION_FILENAME ".gtktermrc"

  Default configuration filename.
- #define BUFFER_LENGTH 256

Functions

- void gtkterm_configuration_default_configuration (GtkTermConfigurationPrivate ∗priv, char ∗section)

  Create a new <default> configuration.
- void gtkterm_configuration_validate (GtkTermConfigurationPrivate ∗priv, char ∗section)

  validate the configuration, given by the section
- bool on_set_config_options (const char ∗name, const char ∗value, gpointer data, GError ∗∗error)

  Set the config option in the keyfile.

Variables

- const char GtkTermConfigurationItems [ ][CONF_ITEM_LENGTH]

  Used configuration options to hold consistency between load/save functions.

5.21.1 Macro Definition Documentation

5.21.1.1 BUFFER_LENGTH

```
#define BUFFER_LENGTH 256
```

5.21.1.2 CONFIGURATION_FILENAME

```
#define CONFIGURATION_FILENAME ".gtktermrc"
```

Default configuration filename.

5.21.2 Function Documentation

5.21.2.1 gtkterm_configuration_default_configuration()

```
void gtkterm_configuration_default_configuration (
            GtkTermConfigurationPrivate * priv,
            char * section )
```

Create a new <default> configuration.

References CONF_ITEM_SERIAL_BAUDRATE, CONF_ITEM_SERIAL_BITS, CONF_ITEM_SERIAL_DISABLE_PORT_LOCK, CONF_ITEM_SERIAL_FLOW_CONTROL, CONF_ITEM_SERIAL_PARITY, CONF_ITEM_SERIAL_PORT, CONF_ITEM_SERIAL_RS485_RTS_TIME_AFTER_TX, CONF_ITEM_SERIAL_RS485_RTS_TIME_BEFORE_TX, CONF_ITEM_SERIAL_STOPBITS, CONF_ITEM_TERM_BACKGROUND_ALPHA, CONF_ITEM_TERM_BACKGROUND_BLUE, CONF_ITEM_TERM_BACKGROUND_GREEN, CONF_ITEM_TERM_BACKGROUND_RED, CONF_ITEM_TERM_BLOCK_CURSO CONF_ITEM_TERM_COLS, CONF_ITEM_TERM_CRLF_AUTO, CONF_ITEM_TERM_ECHO, CONF_ITEM_TERM_FONT, CONF_ITEM_TERM_FOREGROUND_ALPHA, CONF_ITEM_TERM_FOREGROUND_BLUE, CONF_ITEM_TERM_FOREGROUND CONF_ITEM_TERM_FOREGROUND_RED, CONF_ITEM_TERM_MACROS, CONF_ITEM_TERM_ROWS, CONF_ITEM_TERM_SCROLLBACK, CONF_ITEM_TERM_SHOW_CURSOR, CONF_ITEM_TERM_VISUAL_BELL, CONF_ITEM_TERM_WAIT_CHAR, CONF_ITEM_TERM_WAIT_DELAY, DEFAULT_BAUDRATE, DEFAULT_BITS, DEFAULT_CHAR, DEFAULT_DELAY, DEFAULT_DELAY_RS485, DEFAULT_ECHO, DEFAULT_FLOW, DEFAULT_FONT, DEFAULT_PARITY, DEFAULT_PORT, DEFAULT_SCROLLBACK, DEFAULT_STOPBITS, DEFAULT_VISUAL_BELL, GtkTermConfigurationItems, and GtkTermConfigurationPrivate::key_file.

5.21.2.2 gtkterm_configuration_validate()

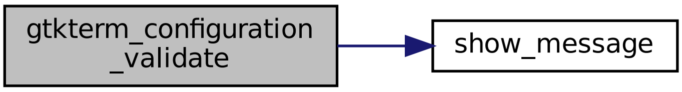
```
void gtkterm_configuration_validate (
            GtkTermConfigurationPrivate * priv,
            char * section )
```

validate the configuration, given by the section

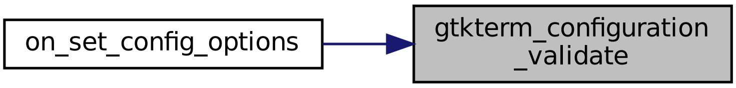
References CONF_ITEM_SERIAL_BAUDRATE, CONF_ITEM_SERIAL_BITS, CONF_ITEM_SERIAL_STOPBITS, CONF_ITEM_TERM_FONT, CONF_ITEM_TERM_WAIT_DELAY, DEFAULT_BITS, DEFAULT_DELAY, DEFAULT_FONT, DEFAULT_STOPBITS, GtkTermConfigurationItems, GtkTermConfigurationPrivate::key_file, MSG_ERR, and show_message().

Referenced by on_set_config_options().

Here is the call graph for this function:



Here is the caller graph for this function:



5.21.2.3 on_set_config_options()

```
bool on_set_config_options (
            const char * name,
            const char * value,
            gpointer data,
            GError ** error )
```

Set the config option in the keyfile.

Options are not saved. Point to the third charater ('–' in front of the cli option)

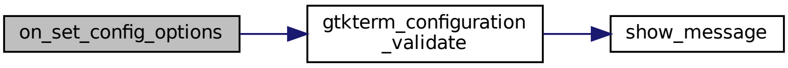
Search index for the option we want to set

Check for max path length. Exit if it is to long. Note: Serial port is also a path to a device.

We should not get here.

References CONF_ITEM_LAST, CONF_ITEM_SERIAL_BAUDRATE, CONF_ITEM_SERIAL_BITS, CONF_ITEM_SERIAL_DISABLE CONF_ITEM_SERIAL_FLOW_CONTROL, CONF_ITEM_SERIAL_PARITY, CONF_ITEM_SERIAL_PORT, CONF_ITEM_SERIAL_RS485_RTS_TIME_AFTER_TX, CONF_ITEM_SERIAL_RS485_RTS_TIME_BEFORE_TX, CONF_ITEM_SERIAL_STOPBITS, CONF_ITEM_TERM_ECHO, CONF_ITEM_TERM_RAW_FILENAME, CONF_ITEM_TERM_WAIT_CHAR, CONF_ITEM_TERM_WAIT_DELAY, config, gtkterm_configuration_validate(), GtkTermConfigurationItems, and GtkTermConfigurationPrivate::key_file.

Here is the call graph for this function:



5.21.3 Variable Documentation

5.21.3.1 GtkTermConfigurationItems

```
const char GtkTermConfigurationItems[ ][CONF_ITEM_LENGTH]
```

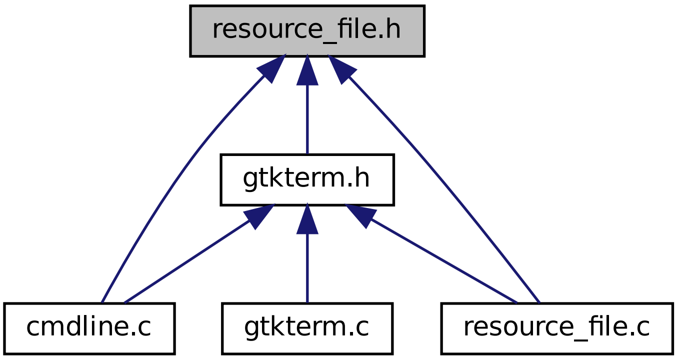
Used configuration options to hold consistency between load/save functions.

Configuration item names.

Referenced by gtkterm_configuration_default_configuration(), gtkterm_configuration_validate(), and on_set_config_options().

5.22 resource_file.h File Reference

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



Classes

- struct _GtkTermConfiguration

Macros

- #define CONF_ITEM_LENGTH 32
- #define DEFAULT_SECTION "default"
- #define GTKTERM_TYPE_CONFIGURATION gtkterm_configuration_get_type ()

Typedefs

- typedef struct _GtkTermConfiguration GtkTermConfiguration

Enumerations

- enum {
  CONF_ITEM_SERIAL_PORT , CONF_ITEM_SERIAL_BAUDRATE , CONF_ITEM_SERIAL_BITS ,
  CONF_ITEM_SERIAL_STOPBITS ,
  CONF_ITEM_SERIAL_PARITY , CONF_ITEM_SERIAL_FLOW_CONTROL , CONF_ITEM_TERM_WAIT_DELAY
  , CONF_ITEM_TERM_WAIT_CHAR ,
  CONF_ITEM_SERIAL_RS485_RTS_TIME_BEFORE_TX , CONF_ITEM_SERIAL_RS485_RTS_TIME_AFTER_TX
  , CONF_ITEM_TERM_MACROS , CONF_ITEM_TERM_RAW_FILENAME ,
  CONF_ITEM_TERM_ECHO , CONF_ITEM_TERM_CRLF_AUTO , CONF_ITEM_SERIAL_DISABLE_PORT_LOCK
  , CONF_ITEM_TERM_FONT ,
  CONF_ITEM_TERM_TIMESTAMP , CONF_ITEM_TERM_BLOCK_CURSOR , CONF_ITEM_TERM_SHOW_CURSOR
  , CONF_ITEM_TERM_ROWS ,
  CONF_ITEM_TERM_COLS , CONF_ITEM_TERM_SCROLLBACK , CONF_ITEM_TERM_VISUAL_BELL ,
  CONF_ITEM_TERM_FOREGROUND_RED ,
  CONF_ITEM_TERM_FOREGROUND_GREEN , CONF_ITEM_TERM_FOREGROUND_BLUE , CONF_ITEM_TERM_FOREG
  , CONF_ITEM_TERM_BACKGROUND_RED ,
  CONF_ITEM_TERM_BACKGROUND_GREEN , CONF_ITEM_TERM_BACKGROUND_BLUE , CONF_ITEM_TERM_BACKG
  , CONF_ITEM_LAST }

  Define all configuration items which are used in the resource file.

Functions

- GtkTermConfiguration ∗ gtkterm_configuration_new (void)
- bool on_set_config_options (const char ∗, const char ∗, gpointer, GError ∗∗)

  Set the config option in the keyfile.

Variables

- const char GtkTermConfigurationItems [ ][CONF_ITEM_LENGTH]

  Configuration item names.

5.22.1 Macro Definition Documentation

5.22.1.1 CONF_ITEM_LENGTH

```
#define CONF_ITEM_LENGTH 32
```

5.22.1.2 DEFAULT_SECTION

```
#define DEFAULT_SECTION "default"
```

5.22.1.3 GTKTERM_TYPE_CONFIGURATION

```
#define GTKTERM_TYPE_CONFIGURATION gtkterm_configuration_get_type ()
```

5.22.2 Typedef Documentation

5.22.2.1 GtkTermConfiguration

```
typedef struct _GtkTermConfiguration GtkTermConfiguration
```

5.22.3 Enumeration Type Documentation

5.22.3.1 anonymous enum

```
anonymous enum
```

Define all configuration items which are used in the resource file.

it is an index to ConfigurationItem.

Enumerator

CONF_ITEM_SERIAL_PORT	
CONF_ITEM_SERIAL_BAUDRATE	
CONF_ITEM_SERIAL_BITS	
CONF_ITEM_SERIAL_STOPBITS	
CONF_ITEM_SERIAL_PARITY	
CONF_ITEM_SERIAL_FLOW_CONTROL	
CONF_ITEM_TERM_WAIT_DELAY	
CONF_ITEM_TERM_WAIT_CHAR	
CONF_ITEM_SERIAL_RS485_RTS_TIME_BEFORE_TX	
CONF_ITEM_SERIAL_RS485_RTS_TIME_AFTER_TX	
CONF_ITEM_TERM_MACROS	
CONF_ITEM_TERM_RAW_FILENAME	
CONF_ITEM_TERM_ECHO	
CONF_ITEM_TERM_CRLF_AUTO	
CONF_ITEM_SERIAL_DISABLE_PORT_LOCK	

Enumerator

CONF_ITEM_TERM_FONT	
CONF_ITEM_TERM_TIMESTAMP	
CONF_ITEM_TERM_BLOCK_CURSOR	
CONF_ITEM_TERM_SHOW_CURSOR	
CONF_ITEM_TERM_ROWS	
CONF_ITEM_TERM_COLS	
CONF_ITEM_TERM_SCROLLBACK	
CONF_ITEM_TERM_VISUAL_BELL	
CONF_ITEM_TERM_FOREGROUND_RED	
CONF_ITEM_TERM_FOREGROUND_GREEN	
CONF_ITEM_TERM_FOREGROUND_BLUE	
CONF_ITEM_TERM_FOREGROUND_ALPHA	
CONF_ITEM_TERM_BACKGROUND_RED	
CONF_ITEM_TERM_BACKGROUND_GREEN	
CONF_ITEM_TERM_BACKGROUND_BLUE	
CONF_ITEM_TERM_BACKGROUND_ALPHA	
CONF_ITEM_LAST	Checking as last item in the list.

5.22.4 Function Documentation

5.22.4.1 gtkterm_configuration_new()

```
GtkTermConfiguration * gtkterm_configuration_new (
             void  )
```

5.22.4.2 on_set_config_options()

```
bool on_set_config_options (
             const char * name,
             const char * value,
             gpointer data,
             GError ** error )
```

Set the config option in the keyfile.

Options are not saved. Point to the third charater ('–' in front of the cli option)

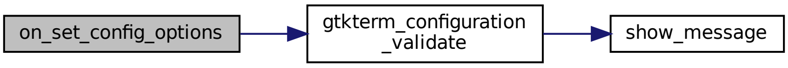
Search index for the option we want to set

Check for max path length. Exit if it is to long. Note: Serial port is also a path to a device.

We should not get here.

References CONF_ITEM_LAST, CONF_ITEM_SERIAL_BAUDRATE, CONF_ITEM_SERIAL_BITS, CONF_ITEM_SERIAL_DISABLE CONF_ITEM_SERIAL_FLOW_CONTROL, CONF_ITEM_SERIAL_PARITY, CONF_ITEM_SERIAL_PORT, CONF_ITEM_SERIAL_RS485_RTS_TIME_AFTER_TX, CONF_ITEM_SERIAL_RS485_RTS_TIME_BEFORE_TX, CONF_ITEM_SERIAL_STOPBITS, CONF_ITEM_TERM_ECHO, CONF_ITEM_TERM_RAW_FILENAME, CONF_ITEM_TERM_WAIT_CHAR, CONF_ITEM_TERM_WAIT_DELAY, config, gtkterm_configuration_validate(), GtkTermConfigurationItems, and GtkTermConfigurationPrivate::key_file.

Here is the call graph for this function:



5.22.5 Variable Documentation

5.22.5.1 GtkTermConfigurationItems

```
const char GtkTermConfigurationItems[][CONF_ITEM_LENGTH]  [extern]
```

Configuration item names.

Configuration item names.

Referenced by gtkterm_configuration_default_configuration(), gtkterm_configuration_validate(), and on_set_config_options().

5.23 resource_file.h

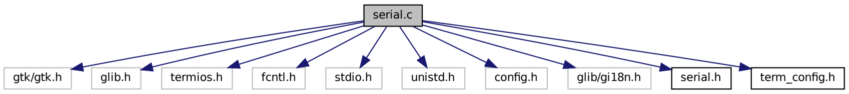
Go to the documentation of this file.

```
/***********************************************************************
* resource_file.h
* ---------------
*           GTKTerm Software
*                      (c) Julien Schmitt
*
* -------------------------------------------------------------------
*
*   \brief Purpose
*      Load and save configuration file
*      - Header file -
*
***********************************************************************/

#ifndef RESOURCE_FILE_H_
#define RESOURCE_FILE_H_

#define CONF_ITEM_LENGTH        32
#define DEFAULT_SECTION        "default"        //!  Default section if not specified

//!  Define all configuration items which are used
//!  in the resource file.  it is an index to ConfigurationItem.
enum {
        CONF_ITEM_SERIAL_PORT,
        CONF_ITEM_SERIAL_BAUDRATE,
        CONF_ITEM_SERIAL_BITS,
        CONF_ITEM_SERIAL_STOPBITS,
        CONF_ITEM_SERIAL_PARITY,
        CONF_ITEM_SERIAL_FLOW_CONTROL,
        CONF_ITEM_TERM_WAIT_DELAY,
        CONF_ITEM_TERM_WAIT_CHAR,
        CONF_ITEM_SERIAL_RS485_RTS_TIME_BEFORE_TX,
        CONF_ITEM_SERIAL_RS485_RTS_TIME_AFTER_TX,
        CONF_ITEM_TERM_MACROS,
        CONF_ITEM_TERM_RAW_FILENAME,
        CONF_ITEM_TERM_ECHO,
        CONF_ITEM_TERM_CRLF_AUTO,
        CONF_ITEM_SERIAL_DISABLE_PORT_LOCK,
        CONF_ITEM_TERM_FONT,
        CONF_ITEM_TERM_TIMESTAMP,
        CONF_ITEM_TERM_BLOCK_CURSOR,
        CONF_ITEM_TERM_SHOW_CURSOR,
        CONF_ITEM_TERM_ROWS,
        CONF_ITEM_TERM_COLS,
        CONF_ITEM_TERM_SCROLLBACK,
        CONF_ITEM_TERM_VISUAL_BELL,
        CONF_ITEM_TERM_FOREGROUND_RED,
        CONF_ITEM_TERM_FOREGROUND_GREEN,
        CONF_ITEM_TERM_FOREGROUND_BLUE,
        CONF_ITEM_TERM_FOREGROUND_ALPHA,
        CONF_ITEM_TERM_BACKGROUND_RED,
        CONF_ITEM_TERM_BACKGROUND_GREEN,
        CONF_ITEM_TERM_BACKGROUND_BLUE,
        CONF_ITEM_TERM_BACKGROUND_ALPHA,
        CONF_ITEM_LAST                        //!  Checking as last item in the list.
};

//!Configuration item names.
extern const char GtkTermConfigurationItems [][CONF_ITEM_LENGTH];

G_BEGIN_DECLS

struct _GtkTermConfiguration {

    GObject parent_instance;
};

#define GTKTERM_TYPE_CONFIGURATION gtkterm_configuration_get_type ()
G_DECLARE_FINAL_TYPE (GtkTermConfiguration, gtkterm_configuration, GTKTERM, CONFIGURATION, GObject)
typedef struct _GtkTermConfiguration GtkTermConfiguration;

GtkTermConfiguration *gtkterm_configuration_new (void);

bool on_set_config_options (const char *, const char *, gpointer,  GError **);

G_END_DECLS

#endif
```

5.24 serial.c File Reference

```
#include <gtk/gtk.h>
#include <glib.h>
#include <termios.h>
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include <config.h>
#include <glib/gi18n.h>
#include "serial.h"
#include "term_config.h"
```

Include dependency graph for serial.c:



Functions

- char ∗ get_port_string (void)

Variables

- port_config_t port_conf
- struct termios termios_save
- int serial_port_fd = -1

5.24.1 Function Documentation

5.24.1.1 get_port_string()

```
char ∗ get_port_string (
            void )
```

References port_config_t::baudrate, port_config_t::bits, port_config_t::parity, port_config_t::port, port_conf, serial_port_fd, and port_config_t::stopbits.

Referenced by set_window_title().

Here is the caller graph for this function:

set_window_title
get_port_string

5.24.2 Variable Documentation

5.24.2.1 port_conf

```
port_config_t port_conf
```

Referenced by get_port_string().

5.24.2.2 serial_port_fd

```
int serial_port_fd = -1
```

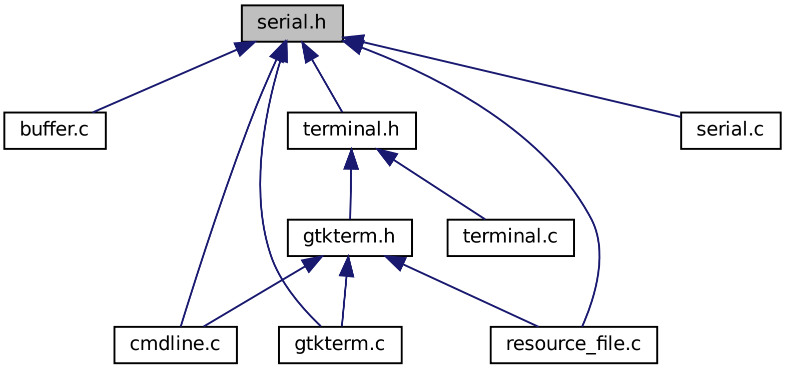
Referenced by get_port_string().

5.24.2.3 termios_save

```
struct termios termios_save
```

5.25 serial.h File Reference

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



Classes

- struct port_config_t

Macros

- #define DEFAULT_PORT "/dev/ttyS0"
- #define DEFAULT_BAUDRATE 115200
- #define DEFAULT_PARITY "none"
- #define DEFAULT_BITS 8
- #define DEFAULT_STOPBITS 1
- #define DEFAULT_FLOW "none"
- #define RECEIVE_BUFFER 8192
- #define TRANSMIT_BUFFER 4096
- #define LINE_FEED 0x0A
- #define POLL_DELAY 100 /∗ in ms (for control signals) ∗/

Functions

- char ∗ get_port_string (void)

Variables

- int serial_port_fd
- port_config_t port_conf

5.25.1 Macro Definition Documentation

5.25.1.1 DEFAULT_BAUDRATE

```
#define DEFAULT_BAUDRATE 115200
```

5.25.1.2 DEFAULT_BITS

```
#define DEFAULT_BITS 8
```

5.25.1.3 DEFAULT_FLOW

```
#define DEFAULT_FLOW "none"
```

5.25.1.4 DEFAULT_PARITY

```
#define DEFAULT_PARITY "none"
```

5.25.1.5 DEFAULT_PORT

```
#define DEFAULT_PORT "/dev/ttyS0"
```

5.25.1.6 DEFAULT_STOPBITS

```
#define DEFAULT_STOPBITS 1
```

5.25.1.7 LINE_FEED

```
#define LINE_FEED 0x0A
```

5.25.1.8 POLL_DELAY

```
#define POLL_DELAY 100 /* in ms (for control signals) */
```

5.25.1.9 RECEIVE_BUFFER

```
#define RECEIVE_BUFFER 8192
```

5.25.1.10 TRANSMIT_BUFFER

```
#define TRANSMIT_BUFFER 4096
```

5.25.2 Function Documentation

5.25.2.1 get_port_string()

```
char * get_port_string (
              void  )
```

References port_config_t::baudrate, port_config_t::bits, port_config_t::parity, port_config_t::port, port_conf, serial_port_fd, and port_config_t::stopbits.

Referenced by set_window_title().

Here is the caller graph for this function:

set_window_title
get_port_string

5.25.3 Variable Documentation

5.25.3.1 port_conf

```
port_config_t port_conf  [extern]
```

Referenced by get_port_string().

5.25.3.2 serial_port_fd

```
int serial_port_fd  [extern]
```

Referenced by get_port_string().

5.26 serial.h

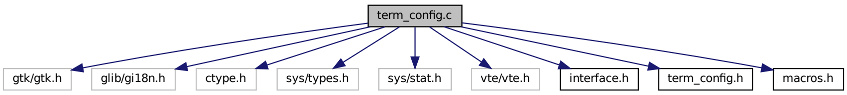
Go to the documentation of this file.

```
/***********************************************************************/
/* serial.h                                                            */
/* -------                                                             */
/*           GTKTerm Software                                          */
/*                      (c) Julien Schmitt                             */
/*                                                                     */
/* ------------------------------------------------------------------- */
/*                                                                     */
/*   Purpose                                                           */
/*      Serial port access functions                                   */
/*      - Header file -                                                */
/*                                                                     */
/***********************************************************************/

#ifndef SERIAL_H_
#define SERIAL_H_

#define DEFAULT_PORT     "/dev/ttyS0"
#define DEFAULT_BAUDRATE 115200
#define DEFAULT_PARITY   "none"
#define DEFAULT_BITS     8
#define DEFAULT_STOPBITS 1
#define DEFAULT_FLOW     "none"

typedef struct
{
    char port[256];
    long int baudrate;              // 300 - 600 - 1200 - ...  - 2000000
    int bits;                       // 5 - 6 - 7 - 8
    int stopbits;                   // 1 - 2
    int parity;                   // 0 :  None, 1 :  Odd, 2 :  Even
    int flow_control;               // 0 :  None, 1 :  Xon/Xoff, 2 :  RTS/CTS, 3 :  RS485halfduplex
    int rs485_rts_time_before_transmit;
    int rs485_rts_time_after_transmit;
    char char_queue;              // character in queue
    bool disable_port_lock;

} port_config_t;

extern int serial_port_fd;
extern port_config_t port_conf;

char* get_port_string (void);

#define RECEIVE_BUFFER 8192
#define TRANSMIT_BUFFER 4096
#define LINE_FEED 0x0A
#define POLL_DELAY 100               /* in ms (for control signals) */

#endif
```

5.27 term_config.c File Reference

```
#include <gtk/gtk.h>
#include <glib/gi18n.h>
#include <ctype.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <vte/vte.h>
#include "interface.h"
#include "term_config.h"
#include "macros.h"
```

Include dependency graph for term_config.c:



Variables

- display_config_t term_conf

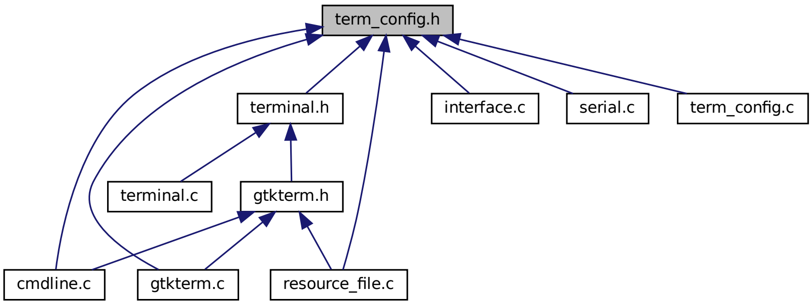
5.27.1 Variable Documentation

5.27.1.1 term_conf

```
display_config_t term_conf
```

5.28 term_config.h File Reference

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



Classes

- struct display_config_t

Macros

- #define DEFAULT_FONT "Monospace 12"
- #define DEFAULT_SCROLLBACK 10000
- #define DEFAULT_DELAY 0
- #define DEFAULT_CHAR -1
- #define DEFAULT_DELAY_RS485 30
- #define DEFAULT_ECHO "false"
- #define DEFAULT_VISUAL_BELL "false"

Variables

- display_config_t term_conf

5.28.1 Macro Definition Documentation

5.28.1.1 DEFAULT_CHAR

```
#define DEFAULT_CHAR -1
```

5.28.1.2 DEFAULT_DELAY

```
#define DEFAULT_DELAY 0
```

5.28.1.3 DEFAULT_DELAY_RS485

```
#define DEFAULT_DELAY_RS485 30
```

5.28.1.4 DEFAULT_ECHO

```
#define DEFAULT_ECHO "false"
```

5.28.1.5 DEFAULT_FONT

```
#define DEFAULT_FONT "Monospace 12"
```

5.28.1.6 DEFAULT_SCROLLBACK

```
#define DEFAULT_SCROLLBACK 10000
```

5.28.1.7 DEFAULT_VISUAL_BELL

```
#define DEFAULT_VISUAL_BELL "false"
```

5.28.2 Variable Documentation

5.28.2.1 term_conf

```
display_config_t term_conf [extern]
```

5.29 term_config.h

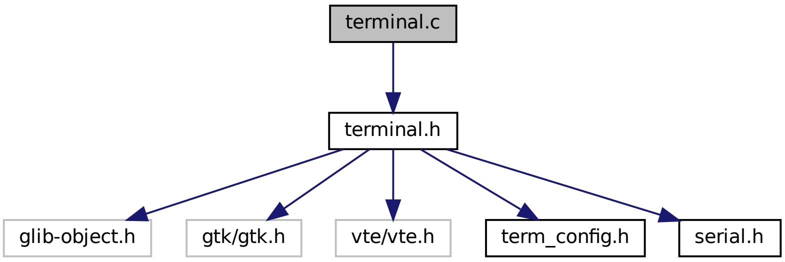
Go to the documentation of this file.

```
/***********************************************************************/
/* term_config.h                                                       */
/* --------                                                            */
/*           GTKTerm Software                                          */
/*                      (c) Julien Schmitt                             */
/*                                                                     */
/* ------------------------------------------------------------------- */
/*                                                                     */
/*   Purpose                                                           */
/*      Configuration of the serial port                               */
/*      - Header file -                                                */
/*                                                                     */
/***********************************************************************/

#ifndef TERM_CONFIG_H_
#define TERM_CONFIG_H_

#define DEFAULT_FONT "Monospace 12"
#define DEFAULT_SCROLLBACK 10000

#define DEFAULT_DELAY       0
#define DEFAULT_CHAR        -1
#define DEFAULT_DELAY_RS485 30
#define DEFAULT_ECHO        "false"
#define DEFAULT_VISUAL_BELL "false"

typedef struct
{
    bool block_cursor;
    bool show_cursor;
    char char_queue;             // character in queue
    bool echo;               // echo local
    bool crlfauto;           // line feed auto
    bool timestamp;
    int delay;                  // end of char delay:  in ms
    int rows;
    int columns;
    int scrollback;
    bool visual_bell;
    GdkRGBA foreground_color;
    GdkRGBA background_color;
    PangoFontDescription *font;
    char *active_section;

    char *default_filename;

} display_config_t;

// configuration for the terminal window
extern display_config_t term_conf;

#endif
```

5.30 terminal.c File Reference

```
#include "terminal.h"
```

Include dependency graph for terminal.c:



Classes

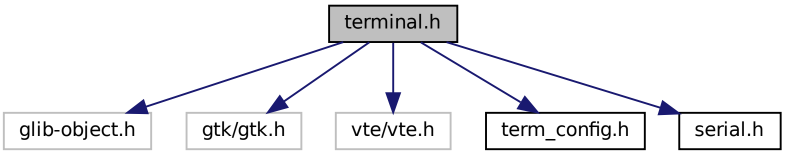
- struct _GtkTermTerminalClass

5.31 terminal.h File Reference

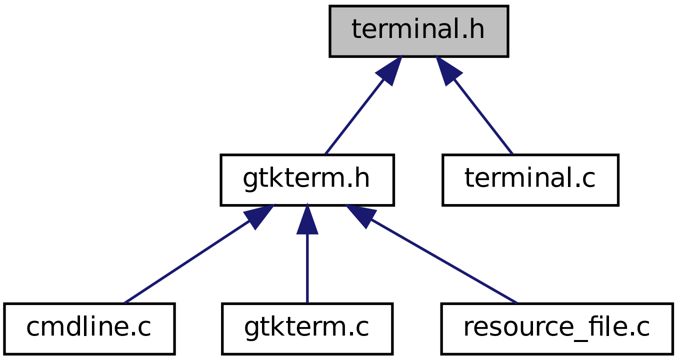
```
#include <glib-object.h>
#include <gtk/gtk.h>
#include <vte/vte.h>
#include <term_config.h>
```

```
#include <serial.h>
```

Include dependency graph for terminal.h:



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



Classes

- struct _GtkTermTerminal

Macros

- #define GTKTERM_TERMINAL_TYPE gtkterm_terminal_get_type()

5.31.1 Macro Definition Documentation

5.31.1.1 GTKTERM_TERMINAL_TYPE

```
#define GTKTERM_TERMINAL_TYPE gtkterm_terminal_get_type()
```

5.32 terminal.h

Go to the documentation of this file.

```
/***********************************************************************/
/* terminal.h                                                        */
/* --------                                                            */
/*           GTKTerm Software                                          */
/*                      (c) Julien Schmitt                             */
/*                                                                     */
/* ------------------------------------------------------------------- */
/*                                                                     */
/*   Purpose                                                           */
/*      Handles all VTE in/output to/from serial port                  */
/*      - Header file -                                                */
/*                                                                     */
/***********************************************************************/
#ifndef TERMINAL_H
#define TERMINAL_H

#include <glib-object.h>
#include <gtk/gtk.h>
#include <vte/vte.h>

#include <term_config.h>
#include <serial.h>

G_BEGIN_DECLS

struct _GtkTermTerminal {
    uint8_t view_mode;              //!  ASCII or HEX view mode
 //   GtkTermBuffer *term_buffer;
    display_config_t *term_conf;    //!  The configuration loaded from the keyfile
    port_config_t *port_conf;       //!  Port configuration used in this terminal
    char *active_section;           //!  Active section in this window from config file

    char *filename;                 //!  File to send

    GdkRGBA *term_forground;        //!  Foreground (text) color of this terminal
    GdkRGBA *term_background;       //!  Background color

    VteTerminal vte_object;         //!  The actual terminal
};


#define GTKTERM_TERMINAL_TYPE gtkterm_terminal_get_type()
G_DECLARE_FINAL_TYPE (GtkTermTerminal, gtkterm_terminal, GTKTERM, TERMINAL, VteTerminal)

G_END_DECLS

#endif // TERMINAL_H
```

Index