

gtkterm

1.99.0

Generated by Doxygen 1.9.1

1 GTKTerm: A GTK+ Serial Port Terminal	1
1.1 Usage	1
1.1.1 Keyboard Shortcuts	1
1.1.2 Command Line Options	1
1.1.3 Notes on RS485:	2
1.1.4 Scriptability with Signals	2
1.2 Installation	2
1.3 Uninstallation	3
1.4 License	3
2 Class Index	5
2.1 Class List	5
3 File Index	7
3.1 File List	7
4 Class Documentation	9
4.1 display_config_t Struct Reference	9
4.1.1 Member Data Documentation	10
4.1.1.1 active_section	10
4.1.1.2 background_color	10
4.1.1.3 block_cursor	10
4.1.1.4 char_queue	10
4.1.1.5 columns	11
4.1.1.6 crlfauto	11
4.1.1.7 delay	11
4.1.1.8 echo	11
4.1.1.9 font	11
4.1.1.10 foreground_color	11
4.1.1.11 rows	12
4.1.1.12 scrollbar	12
4.1.1.13 show_cursor	12
4.1.1.14 timestamp	12
4.1.1.15 visual_bell	12
4.2 GtkTermWindow Struct Reference	13
4.2.1 Member Data Documentation	13
4.2.1.1 buffer	13
4.2.1.2 fullscreen	14
4.2.1.3 height	14
4.2.1.4 infobar	14
4.2.1.5 maximized	14
4.2.1.6 menubutton	14
4.2.1.7 message	14

4.2.1.8 parent_instance	14
4.2.1.9 scrolled_window	14
4.2.1.10 status	15
4.2.1.11 toolmenu	15
4.2.1.12 width	15
4.3 macro_t Struct Reference	15
4.3.1 Detailed Description	16
4.3.2 Member Data Documentation	16
4.3.2.1 action	16
4.3.2.2 closure	16
4.3.2.3 shortcut	16
4.4 port_config_t Struct Reference	17
4.4.1 Member Data Documentation	17
4.4.1.1 bits	17
4.4.1.2 char_queue	18
4.4.1.3 disable_port_lock	18
4.4.1.4 flow_control	18
4.4.1.5 parity	18
4.4.1.6 port	18
4.4.1.7 rs485_rts_time_after_transmit	18
4.4.1.8 rs485_rts_time_before_transmit	19
4.4.1.9 speed	19
4.4.1.10 stops	19
5 File Documentation	21
5.1 buffer.c File Reference	21
5.1.1 Macro Definition Documentation	22
5.1.1.1 TIMESTAMP_SIZE	22
5.1.2 Function Documentation	22
5.1.2.1 clear_buffer()	22
5.1.2.2 create_buffer()	22
5.1.2.3 delete_buffer()	22
5.1.2.4 insert_timestamp()	23
5.1.2.5 put_chars()	23
5.1.2.6 set_clear_func()	23
5.1.2.7 set_display_func()	23
5.1.2.8 unset_clear_func()	23
5.1.2.9 unset_display_func()	23
5.1.2.10 write_buffer()	24
5.1.2.11 write_buffer_with_func()	24
5.1.3 Variable Documentation	24
5.1.3.1 clear_func	24

5.1.3.2 overlapped	25
5.1.3.3 timestamp_on	25
5.1.3.4 virt_col_pos	25
5.1.3.5 write_func	25
5.2 buffer.h File Reference	25
5.2.1 Macro Definition Documentation	26
5.2.1.1 BUFFER_SIZE	26
5.2.2 Function Documentation	26
5.2.2.1 clear_buffer()	26
5.2.2.2 create_buffer()	26
5.2.2.3 delete_buffer()	27
5.2.2.4 put_chars()	27
5.2.2.5 set_clear_func()	27
5.2.2.6 set_display_func()	27
5.2.2.7 unset_clear_func()	27
5.2.2.8 unset_display_func()	27
5.2.2.9 write_buffer()	28
5.2.2.10 write_buffer_with_func()	28
5.3 cmdline.c File Reference	28
5.3.1 Function Documentation	29
5.3.1.1 display_help()	29
5.3.1.2 read_command_line()	30
5.3.2 Variable Documentation	30
5.3.2.1 config	30
5.4 cmdline.h File Reference	31
5.4.1 Function Documentation	31
5.4.1.1 read_command_line()	31
5.5 files.c File Reference	31
5.5.1 Variable Documentation	32
5.5.1.1 default_filename	32
5.6 files.h File Reference	32
5.6.1 Function Documentation	32
5.6.1.1 add_input()	33
5.6.1.2 save_raw_file()	33
5.6.1.3 send_raw_file()	33
5.6.2 Variable Documentation	33
5.6.2.1 default_filename	33
5.6.2.2 waiting_for_char	33
5.7 gtkterm.c File Reference	34
5.7.1 Typedef Documentation	34
5.7.1.1 GtkTerm	34
5.7.1.2 GtkTermClass	35

5.7.1.3 GtkTermWindowClass	35
5.7.2 Function Documentation	35
5.7.2.1 main()	35
5.7.2.2 set_window_title()	36
5.8 i18n.c File Reference	36
5.8.1 Function Documentation	36
5.8.1.1 i18n_fprintf()	37
5.8.1.2 i18n_perror()	37
5.8.1.3 i18n_printf()	37
5.8.1.4 strerror_utf8()	37
5.9 i18n.h File Reference	38
5.9.1 Macro Definition Documentation	38
5.9.1.1 I18N_H	38
5.9.2 Function Documentation	39
5.9.2.1 i18n_fprintf()	39
5.9.2.2 i18n_perror()	39
5.9.2.3 i18n_printf()	39
5.9.2.4 strerror_utf8()	39
5.10 interface.c File Reference	40
5.10.1 Function Documentation	40
5.10.1.1 show_message()	40
5.10.2 Variable Documentation	41
5.10.2.1 config	41
5.10.2.2 timestamp_on	41
5.10.2.3 virt_col_pos	41
5.11 interface.h File Reference	41
5.11.1 Macro Definition Documentation	42
5.11.1.1 ASCII_VIEW	42
5.11.1.2 HEXADECIMAL_VIEW	42
5.11.1.3 MSG_ERR	42
5.11.1.4 MSG_WRN	42
5.11.2 Function Documentation	42
5.11.2.1 show_message()	42
5.11.3 Variable Documentation	43
5.11.3.1 display	43
5.11.3.2 Text	43
5.12 macros.c File Reference	43
5.12.1 Enumeration Type Documentation	44
5.12.1.1 anonymous enum	44
5.12.2 Function Documentation	44
5.12.2.1 convert_macros_to_string()	44
5.12.2.2 convert_string_to_macros()	45

5.12.2.3 <code>get_shortcuts()</code>	45
5.12.2.4 <code>macro_count()</code>	46
5.12.2.5 <code>remove_shortcuts()</code>	46
5.12.3 Variable Documentation	46
5.12.3.1 <code>macros</code>	46
5.12.3.2 <code>nr_of_macros</code>	47
5.13 <code>macros.h</code> File Reference	47
5.13.1 Function Documentation	47
5.13.1.1 <code>add_shortcuts()</code>	48
5.13.1.2 <code>convert_macros_to_string()</code>	48
5.13.1.3 <code>convert_string_to_macros()</code>	48
5.13.1.4 <code>get_shortcuts()</code>	49
5.13.1.5 <code>macro_count()</code>	49
5.13.1.6 <code>remove_shortcuts()</code>	49
5.13.2 Variable Documentation	50
5.13.2.1 <code>macros</code>	50
5.14 <code>README.md</code> File Reference	50
5.15 <code>README.source</code> File Reference	50
5.16 <code>resource_file.c</code> File Reference	50
5.16.1 Macro Definition Documentation	51
5.16.1.1 <code>CONFIGURATION_FILENAME</code>	52
5.16.1.2 <code>DEFAULT_SECTION</code>	52
5.16.2 Enumeration Type Documentation	52
5.16.2.1 anonymous enum	52
5.16.3 Function Documentation	53
5.16.3.1 <code>check_configuration_file()</code>	53
5.16.3.2 <code>config_file_init()</code>	54
5.16.3.3 <code>copy_configuration()</code>	54
5.16.3.4 <code>dump_configuration_to_cli()</code>	55
5.16.3.5 <code>hard_default_configuration()</code>	56
5.16.3.6 <code>load_configuration_from_file()</code>	57
5.16.3.7 <code>remove_section()</code>	58
5.16.3.8 <code>save_configuration_to_file()</code>	59
5.16.3.9 <code>set_color()</code>	59
5.16.3.10 <code>validate_configuration()</code>	60
5.16.4 Variable Documentation	60
5.16.4.1 <code>config_file</code>	60
5.16.4.2 <code>ConfigurationItem</code>	61
5.17 <code>resource_file.h</code> File Reference	61
5.17.1 Function Documentation	62
5.17.1.1 <code>check_configuration_file()</code>	62
5.17.1.2 <code>config_file_init()</code>	63

5.17.1.3	copy_configuration()	63
5.17.1.4	dump_configuration_to_cli()	64
5.17.1.5	hard_default_configuration()	65
5.17.1.6	load_configuration_from_file()	66
5.17.1.7	remove_section()	67
5.17.1.8	save_configuration_to_file()	68
5.17.1.9	set_color()	69
5.17.1.10	validate_configuration()	69
5.17.2	Variable Documentation	70
5.17.2.1	config_file	70
5.18	serial.c File Reference	70
5.18.1	Function Documentation	70
5.18.1.1	get_port_string()	71
5.18.2	Variable Documentation	71
5.18.2.1	port_conf	71
5.18.2.2	serial_port_fd	71
5.18.2.3	termios_save	71
5.19	serial.h File Reference	72
5.19.1	Macro Definition Documentation	72
5.19.1.1	DEFAULT_BITS	73
5.19.1.2	DEFAULT_FLOW	73
5.19.1.3	DEFAULT_PARITY	73
5.19.1.4	DEFAULT_PORT	73
5.19.1.5	DEFAULT_SPEED	73
5.19.1.6	DEFAULT_STOP	73
5.19.1.7	LINE_FEED	73
5.19.1.8	POLL_DELAY	73
5.19.1.9	RECEIVE_BUFFER	74
5.19.1.10	TRANSMIT_BUFFER	74
5.19.2	Function Documentation	74
5.19.2.1	get_port_string()	74
5.19.3	Variable Documentation	74
5.19.3.1	port_conf	74
5.19.3.2	serial_port_fd	75
5.20	term_config.c File Reference	75
5.20.1	Variable Documentation	75
5.20.1.1	term_conf	75
5.21	term_config.h File Reference	75
5.21.1	Macro Definition Documentation	76
5.21.1.1	DEFAULT_CHAR	76
5.21.1.2	DEFAULT_DELAY	76
5.21.1.3	DEFAULT_DELAY_RS485	76

5.21.1.4 DEFAULT_ECHO	76
5.21.1.5 DEFAULT_FONT	77
5.21.1.6 DEFAULT_SCROLLBACK	77
5.21.2 Variable Documentation	77
5.21.2.1 term_conf	77
Index	79

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
<code><ctrl><shift>L</code>	Clear screen
<code><ctrl><shift>R</code>	Send file
<code><ctrl><shift>Q</code>	Quit
<code><ctrl><shift>S</code>	Configure port
<code><ctrl><shift>V</code>	Paste
<code><ctrl><shift>C</code>	Copy
<code><ctrl><shift>F</code>	Find
<code><ctrl><shift>K</code>	Clear Scrollback
<code><ctrl><shift>A</code>	Select All
<code><ctrl><shift>B</code>	Send Break
<code><ctrl>B</code>	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	<code>killall -USR1 gtkterm</code>
SIGUSR2	Close Port	<code>killall -USR2 gtkterm</code>

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:

display_config_t	9
GtkTermWindow	13
macro_t	
Define macro structure type	15
port_config_t	17

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

buffer.c	21
buffer.h	25
cmdline.c	28
cmdline.h	31
files.c	31
files.h	32
gtkterm.c	34
i18n.c	36
i18n.h	38
interface.c	40
interface.h	41
macros.c	43
macros.h	47
README.source	50
resource_file.c	50
resource_file.h	61
serial.c	70
serial.h	72
term_config.c	75
term_config.h	75

Chapter 4

Class Documentation

4.1 display_config_t Struct Reference

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

Collaboration diagram for display_config_t:

display_config_t
<div><div>+ block_cursor</div><div>+ show_cursor</div><div>+ char_queue</div><div>+ echo</div><div>+ crlfauto</div><div>+ timestamp</div><div>+ delay</div><div>+ rows</div><div>+ columns</div><div>+ scrollbar</div><div>+ visual_bell</div><div>+ foreground_color</div><div>+ background_color</div><div>+ font</div><div>+ active_section</div></div>

Public Attributes

- gboolean [block_cursor](#)
- gboolean [show_cursor](#)
- char [char_queue](#)

- gboolean [echo](#)
- gboolean [crlfauto](#)
- gboolean [timestamp](#)
- int [delay](#)
- int [rows](#)
- int [columns](#)
- int [scrollback](#)
- gboolean [visual_bell](#)
- GdkRGBA [foreground_color](#)
- GdkRGBA [background_color](#)
- PangoFontDescription * [font](#)
- char * [active_section](#)

4.1.1 Member Data Documentation

4.1.1.1 `active_section`

`char* display_config_t::active_section`

Referenced by [check_configuration_file\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), and [remove_section\(\)](#).

4.1.1.2 `background_color`

`GdkRGBA display_config_t::background_color`

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), and [load_configuration_from_file\(\)](#).

4.1.1.3 `block_cursor`

`gboolean display_config_t::block_cursor`

Referenced by [dump_configuration_to_cli\(\)](#), and [hard_default_configuration\(\)](#).

4.1.1.4 `char_queue`

`char display_config_t::char_queue`

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), and [read_command_line\(\)](#).

4.1.1.5 columns

```
int display_config_t::columns
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), and [load_configuration_from_file\(\)](#).

4.1.1.6 crlfauto

```
gboolean display_config_t::crlfauto
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), and [load_configuration_from_file\(\)](#).

4.1.1.7 delay

```
int display_config_t::delay
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), [read_command_line\(\)](#), and [validate_configuration\(\)](#).

4.1.1.8 echo

```
gboolean display_config_t::echo
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), and [read_command_line\(\)](#).

4.1.1.9 font

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

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), and [validate_configuration\(\)](#).

4.1.1.10 foreground_color

```
GdkRGBA display_config_t::foreground_color
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), and [load_configuration_from_file\(\)](#).

4.1.1.11 rows

```
int display_config_t::rows
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), and [load_configuration_from_file\(\)](#).

4.1.1.12 scrollbar

```
int display_config_t::scrollback
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), and [load_configuration_from_file\(\)](#).

4.1.1.13 show_cursor

```
gboolean display_config_t::show_cursor
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), and [load_configuration_from_file\(\)](#).

4.1.1.14 timestamp

```
gboolean display_config_t::timestamp
```

Referenced by [dump_configuration_to_cli\(\)](#), and [hard_default_configuration\(\)](#).

4.1.1.15 visual_bell

```
gboolean display_config_t::visual_bell
```

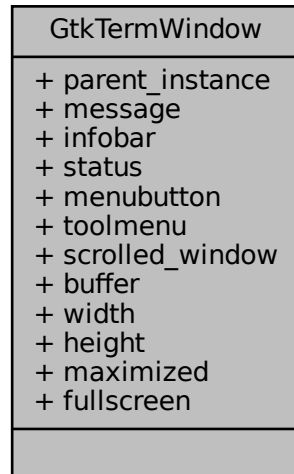
Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), and [load_configuration_from_file\(\)](#).

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

- [term_config.h](#)

4.2 GtkTermWindow Struct Reference

Collaboration diagram for GtkTermWindow:



Public Attributes

- `GtkApplicationWindow` [parent_instance](#)
- `GtkWidget` * [message](#)
- `GtkWidget` * [infobar](#)
- `GtkWidget` * [status](#)
- `GtkWidget` * [menubutton](#)
- `GMenuModel` * [toolmenu](#)
- `GtkScrolledWindow` * [scrolled_window](#)
- `GtkWidget` * [buffer](#)
- `int` [width](#)
- `int` [height](#)
- `gboolean` [maximized](#)
- `gboolean` [fullscreen](#)

4.2.1 Member Data Documentation

4.2.1.1 buffer

`GtkWidget*` `GtkTermWindow::buffer`

4.2.1.2 fullscreen

`gboolean GtkTermWindow::fullscreen`

4.2.1.3 height

`int GtkTermWindow::height`

4.2.1.4 infobar

`GtkWidget* GtkTermWindow::infobar`

4.2.1.5 maximized

`gboolean GtkTermWindow::maximized`

4.2.1.6 menubutton

`GtkWidget* GtkTermWindow::menubutton`

4.2.1.7 message

`GtkWidget* GtkTermWindow::message`

4.2.1.8 parent_instance

`GtkApplicationWindow GtkTermWindow::parent_instance`

4.2.1.9 scrolled_window

`GtkScrolledWindow* GtkTermWindow::scrolled_window`

4.2.1.10 status

```
GtkWidget* GtkTermWindow::status
```

4.2.1.11 toolmenu

```
GMenuModel* GtkTermWindow::toolmenu
```

4.2.1.12 width

```
int GtkTermWindow::width
```

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

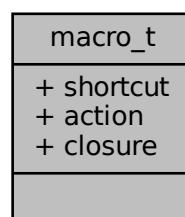
- [gtkterm.c](#)

4.3 macro_t Struct Reference

Define macro structure type.

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

Collaboration diagram for macro_t:



Public Attributes

- char * [shortcut](#)
- char * [action](#)
Shortcut of the macro.
- GClosure * [closure](#)
Command to perform.

4.3.1 Detailed Description

Define macro structure type.

4.3.2 Member Data Documentation

4.3.2.1 action

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

Shortcut of the macro.

Referenced by [convert_macros_to_string\(\)](#), and [convert_string_to_macros\(\)](#).

4.3.2.2 closure

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

Command to perform.

4.3.2.3 shortcut

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

Referenced by [convert_macros_to_string\(\)](#), and [convert_string_to_macros\(\)](#).

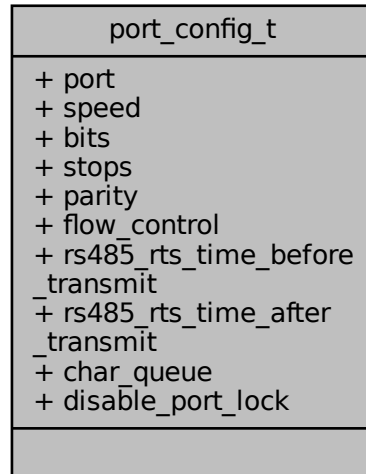
The documentation for this struct was generated from the following file:

- [macros.h](#)

4.4 port_config_t Struct Reference

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

Collaboration diagram for port_config_t:



Public Attributes

- char [port](#) [256]
- long int [speed](#)
- int [bits](#)
- int [stops](#)
- int [parity](#)
- int [flow_control](#)
- int [rs485_rts_time_before_transmit](#)
- int [rs485_rts_time_after_transmit](#)
- char [char_queue](#)
- gboolean [disable_port_lock](#)

4.4.1 Member Data Documentation

4.4.1.1 bits

```
int port_config_t::bits
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [get_port_string\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), [read_command_line\(\)](#), and [validate_configuration\(\)](#).

4.4.1.2 char_queue

```
char port_config_t::char_queue
```

4.4.1.3 disable_port_lock

```
gboolean port_config_t::disable_port_lock
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), and [read_command_line\(\)](#).

4.4.1.4 flow_control

```
int port_config_t::flow_control
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), and [read_command_line\(\)](#).

4.4.1.5 parity

```
int port_config_t::parity
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [get_port_string\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), and [read_command_line\(\)](#).

4.4.1.6 port

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

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [get_port_string\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), and [read_command_line\(\)](#).

4.4.1.7 rs485_rts_time_after_transmit

```
int port_config_t::rs485_rts_time_after_transmit
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), and [read_command_line\(\)](#).

4.4.1.8 rs485_rts_time_before_transmit

```
int port_config_t::rs485_rts_time_before_transmit
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), and [read_command_line\(\)](#).

4.4.1.9 speed

```
long int port_config_t::speed
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [get_port_string\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), [read_command_line\(\)](#), and [validate_configuration\(\)](#).

4.4.1.10 stops

```
int port_config_t::stops
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [get_port_string\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), [read_command_line\(\)](#), and [validate_configuration\(\)](#).

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

- [serial.h](#)

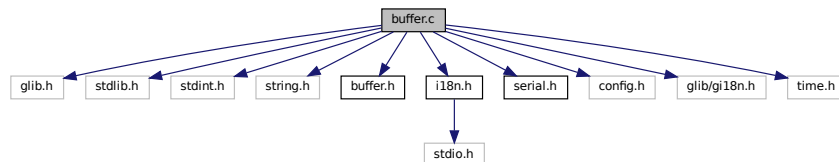
Chapter 5

File Documentation

5.1 buffer.c File Reference

```
#include <glib.h>
#include <stdlib.h>
#include <stdint.h>
#include <string.h>
#include "buffer.h"
#include "i18n.h"
#include "serial.h"
#include <config.h>
#include <glib/gi18n.h>
#include <time.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, gboolean 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

- gboolean [timestamp_on](#)
- char [overlapped](#)
- guint [virt_col_pos](#)
- void(* [write_func](#))(const char *, unsigned int) = NULL
- void(* [clear_func](#))(void) = NULL

5.1.1 Macro Definition Documentation

5.1.1.1 TIMESTAMP_SIZE

```
#define TIMESTAMP_SIZE 50
```

5.1.2 Function Documentation

5.1.2.1 clear_buffer()

```
void clear_buffer (  
    void )
```

References [clear_func](#).

5.1.2.2 create_buffer()

```
void create_buffer (  
    void )
```

5.1.2.3 delete_buffer()

```
void delete_buffer (  
    void )
```


5.1.2.4 insert_timestamp()

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

5.1.2.5 put_chars()

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

References [RECEIVE_BUFFER](#), [timestamp_on](#), and [TIMESTAMP_SIZE](#).

5.1.2.6 set_clear_func()

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

References [clear_func](#).

5.1.2.7 set_display_func()

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

References [write_func](#).

5.1.2.8 unset_clear_func()

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

References [clear_func](#).

5.1.2.9 unset_display_func()

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

References [write_func](#).

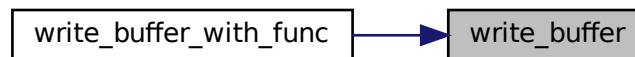
5.1.2.10 write_buffer()

```
void write_buffer (
    void )
```

References [overlapped](#), and [write_func](#).

Referenced by [write_buffer_with_func\(\)](#).

Here is the caller graph for this function:

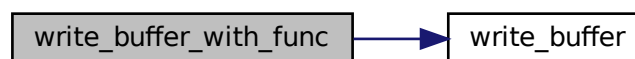


5.1.2.11 write_buffer_with_func()

```
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:



5.1.3 Variable Documentation

5.1.3.1 clear_func

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

Referenced by [clear_buffer\(\)](#), [set_clear_func\(\)](#), and [unset_clear_func\(\)](#).

5.1.3.2 overlapped

```
char overlapped
```

Referenced by [write_buffer\(\)](#).

5.1.3.3 timestamp_on

```
gboolean timestamp_on [extern]
```

Referenced by [put_chars\(\)](#).

5.1.3.4 virt_col_pos

```
guint virt_col_pos [extern]
```

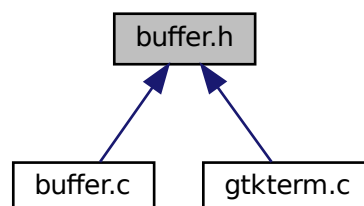
5.1.3.5 write_func

```
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\(\)](#).

5.2 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, gboolean)
- 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))

5.2.1 Macro Definition Documentation

5.2.1.1 BUFFER_SIZE

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

5.2.2 Function Documentation

5.2.2.1 clear_buffer()

```
void clear_buffer (  
    void )
```

References [clear_func](#).

5.2.2.2 create_buffer()

```
void create_buffer (  
    void )
```

5.2.2.3 delete_buffer()

```
void delete_buffer (
    void )
```

5.2.2.4 put_chars()

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

References [RECEIVE_BUFFER](#), [timestamp_on](#), and [TIMESTAMP_SIZE](#).

5.2.2.5 set_clear_func()

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

References [clear_func](#).

5.2.2.6 set_display_func()

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

5.2.2.7 unset_clear_func()

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

References [clear_func](#).

5.2.2.8 unset_display_func()

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

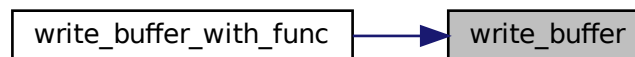
5.2.2.9 write_buffer()

```
void write_buffer (
    void )
```

References [overlapped](#), and [write_func](#).

Referenced by [write_buffer_with_func\(\)](#).

Here is the caller graph for this function:



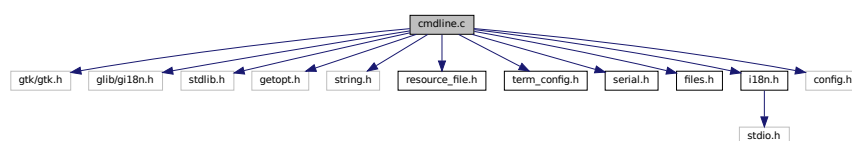
5.2.2.10 write_buffer_with_func()

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

5.3 cmdline.c File Reference

```
#include <gtk/gtk.h>
#include <glib/gi18n.h>
#include <stdlib.h>
#include <getopt.h>
#include <string.h>
#include "resource_file.h"
#include "term_config.h"
#include "serial.h"
#include "files.h"
#include "i18n.h"
#include <config.h>
```

Include dependency graph for cmdline.c:



Functions

- void [display_help](#) (void)
- int [read_command_line](#) (int argc, char **argv, char *configuration_to_read)

Variables

- struct configuration_port [config](#)

5.3.1 Function Documentation

5.3.1.1 [display_help\(\)](#)

```
void display_help (  
    void )
```

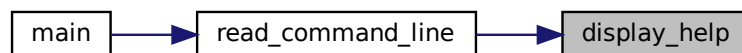
References [i18n_printf\(\)](#).

Referenced by [read_command_line\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



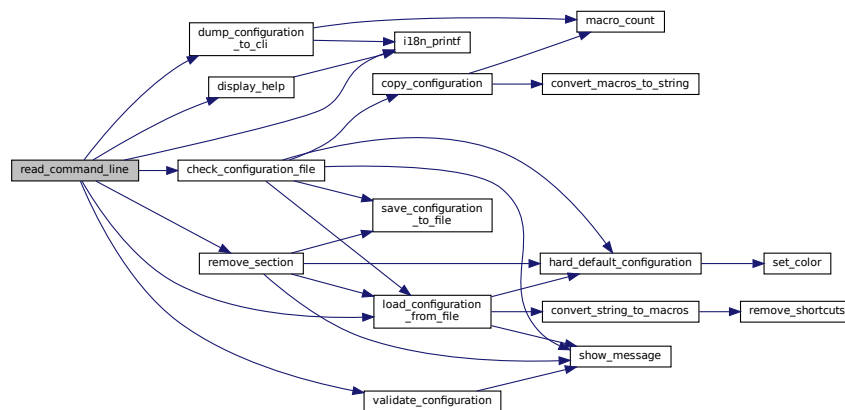
5.3.1.2 read_command_line()

```
int read_command_line (
    int argc,
    char ** argv,
    char * configuration_to_read )
```

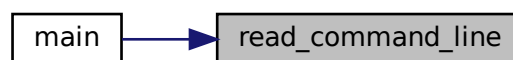
References [port_config_t::bits](#), [display_config_t::char_queue](#), [check_configuration_file\(\)](#), [default_filename](#), [display_config_t::delay](#), [port_config_t::disable_port_lock](#), [display_help\(\)](#), [dump_configuration_to_cli\(\)](#), [display_config_t::echo](#), [port_config_t::flow_control](#), [i18n_printf\(\)](#), [load_configuration_from_file\(\)](#), [port_config_t::parity](#), [port_config_t::port](#), [port_conf](#), [remove_section\(\)](#), [port_config_t::rs485_rts_time_after_transmit](#), [port_config_t::rs485_rts_time_before_transmit](#), [port_config_t::speed](#), [port_config_t::stops](#), [term_conf](#), and [validate_configuration\(\)](#).

Referenced by [main\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.2 Variable Documentation

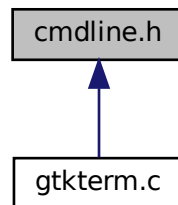
5.3.2.1 config

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

Referenced by [check_configuration_file\(\)](#), [remove_section\(\)](#), and [save_configuration_to_file\(\)](#).

5.4 cmdline.h File Reference

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



Functions

- int [read_command_line](#) (int, char **)

5.4.1 Function Documentation

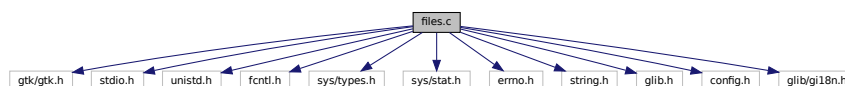
5.4.1.1 read_command_line()

```
int read_command_line (  
    int ,  
    char ** )
```

5.5 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/glib.h>
```

Include dependency graph for files.c:



Variables

- char * [default_filename](#) = NULL

5.5.1 Variable Documentation

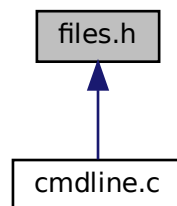
5.5.1.1 default_filename

```
char* default_filename = NULL
```

Referenced by [read_command_line\(\)](#).

5.6 files.h File Reference

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



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](#)

5.6.1 Function Documentation

5.6.1.1 add_input()

```
void add_input (
    void )
```

5.6.1.2 save_raw_file()

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

5.6.1.3 send_raw_file()

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

5.6.2 Variable Documentation

5.6.2.1 default_filename

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

Referenced by [read_command_line\(\)](#).

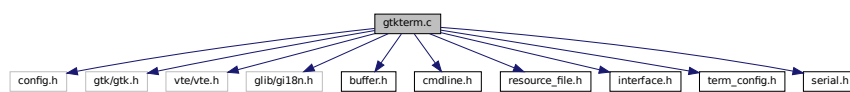
5.6.2.2 waiting_for_char

```
gboolean waiting_for_char [extern]
```

5.7 gtkterm.c File Reference

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

Include dependency graph for gtkterm.c:



Classes

- struct [GtkTermWindow](#)

Typedefs

- typedef GtkApplication [GtkTerm](#)
- typedef GtkApplicationClass [GtkTermClass](#)
- typedef GtkApplicationWindowClass [GtkTermWindowClass](#)

Functions

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

5.7.1 Typedef Documentation

5.7.1.1 GtkTerm

```
typedef GtkApplication GtkTerm
```

5.7.1.2 GtkTermClass

```
typedef GtkApplicationClass GtkTermClass
```

5.7.1.3 GtkTermWindowClass

```
typedef GtkApplicationWindowClass GtkTermWindowClass
```

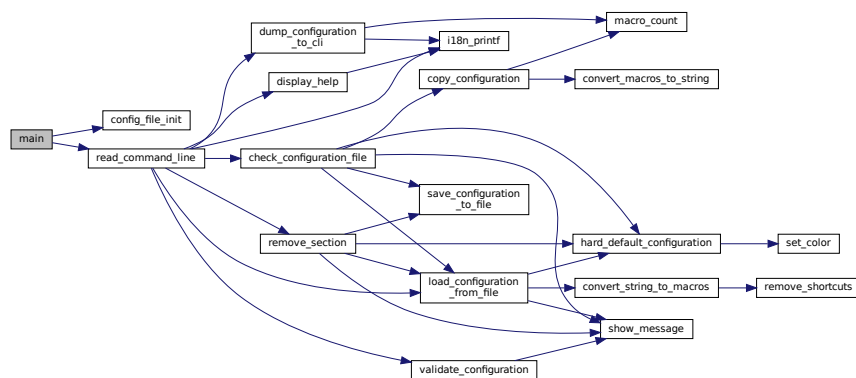
5.7.2 Function Documentation

5.7.2.1 main()

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

References [config_file_init\(\)](#), and [read_command_line\(\)](#).

Here is the call graph for this function:



5.7.2.2 set_window_title()

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

References [get_port_string\(\)](#).

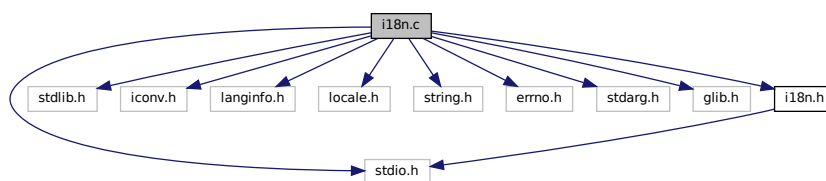
Here is the call graph for this function:



5.8 i18n.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <iconv.h>
#include <langinfo.h>
#include <locale.h>
#include <string.h>
#include <errno.h>
#include <stdarg.h>
#include <glib.h>
#include "i18n.h"
```

Include dependency graph for i18n.c:



Functions

- int [i18n_printf](#) (const char *format,...)
- int [i18n_fprintf](#) (FILE *stream, const char *format,...)
- void [i18n_perror](#) (const char *s)
- char * [strerror_utf8](#) (int errornum)

5.8.1 Function Documentation

5.8.1.1 i18n_fprintf()

```
int i18n_fprintf (
    FILE * stream,
    const char * format,
    ... )
```

5.8.1.2 i18n_perror()

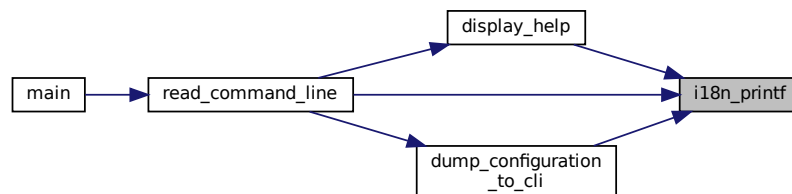
```
void i18n_perror (
    const char * s )
```

5.8.1.3 i18n_printf()

```
int i18n_printf (
    const char * format,
    ... )
```

Referenced by [display_help\(\)](#), [dump_configuration_to_cli\(\)](#), and [read_command_line\(\)](#).

Here is the caller graph for this function:



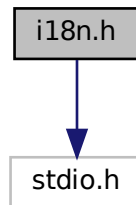
5.8.1.4 strerror_utf8()

```
char* strerror_utf8 (
    int errornum )
```

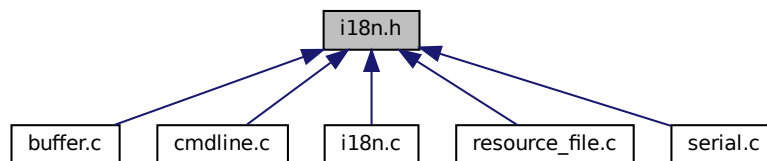
5.9 i18n.h File Reference

```
#include <stdio.h>
```

Include dependency graph for i18n.h:



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



Macros

- `#define I18N_H`

Functions

- `int i18n_printf (const char *,...)`
- `int i18n_fprintf (FILE *, const char *,...)`
- `void i18n_perror (const char *)`
- `char * strerror_utf8 (int)`

5.9.1 Macro Definition Documentation

5.9.1.1 I18N_H

```
#define I18N_H
```


5.9.2 Function Documentation

5.9.2.1 i18n_fprintf()

```
int i18n_fprintf (
    FILE * stream,
    const char * format,
    ... )
```

5.9.2.2 i18n_perror()

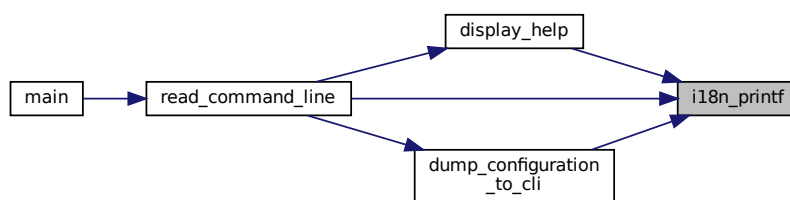
```
void i18n_perror (
    const char * s )
```

5.9.2.3 i18n_printf()

```
int i18n_printf (
    const char * format,
    ... )
```

Referenced by [display_help\(\)](#), [dump_configuration_to_cli\(\)](#), and [read_command_line\(\)](#).

Here is the caller graph for this function:

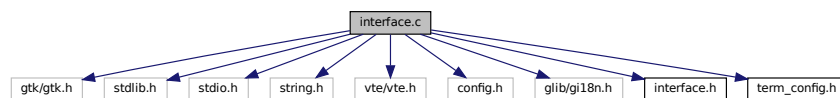


5.9.2.4 strerror_utf8()

```
char* strerror_utf8 (
    int errornum )
```

5.10 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/glib.h>
#include "interface.h"
#include "term_config.h"
Include dependency graph for interface.c:
```



Functions

- void [show_message](#) (char *message, int type_msg)

Variables

- gboolean [timestamp_on](#) = 0
- struct configuration_port [config](#)
- int [virt_col_pos](#) = 0

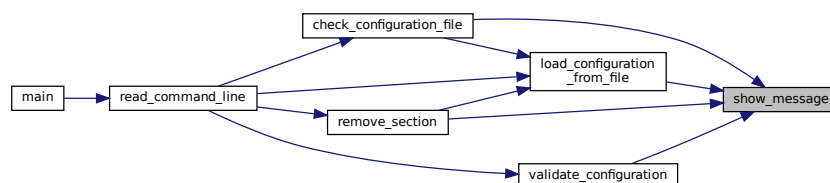
5.10.1 Function Documentation

5.10.1.1 show_message()

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

Referenced by [check_configuration_file\(\)](#), [load_configuration_from_file\(\)](#), [remove_section\(\)](#), and [validate_configuration\(\)](#).

Here is the caller graph for this function:



5.10.2 Variable Documentation

5.10.2.1 config

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

5.10.2.2 timestamp_on

```
gboolean timestamp_on = 0
```

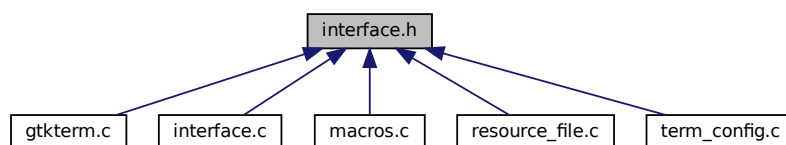
Referenced by [put_chars\(\)](#).

5.10.2.3 virt_col_pos

```
int virt_col_pos = 0
```

5.11 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](#)

5.11.1 Macro Definition Documentation

5.11.1.1 ASCII_VIEW

```
#define ASCII_VIEW 0
```

5.11.1.2 HEXADECIMAL_VIEW

```
#define HEXADECIMAL_VIEW 1
```

5.11.1.3 MSG_ERR

```
#define MSG_ERR 1
```

5.11.1.4 MSG_WRN

```
#define MSG_WRN 0
```

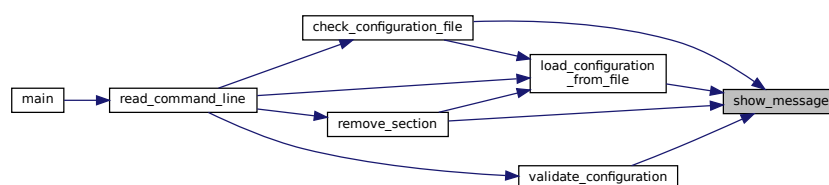
5.11.2 Function Documentation

5.11.2.1 show_message()

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

Referenced by [check_configuration_file\(\)](#), [load_configuration_from_file\(\)](#), [remove_section\(\)](#), and [validate_configuration\(\)](#).

Here is the caller graph for this function:



5.11.3 Variable Documentation

5.11.3.1 display

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

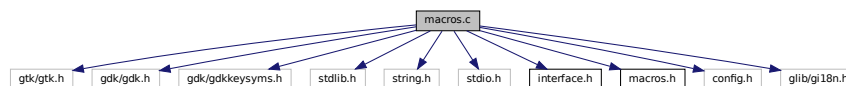
5.11.3.2 Text

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

5.12 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/glib.h>
```

Include dependency graph for macros.c:



Enumerations

- enum { COLUMN_SHORTCUT , COLUMN_ACTION , NUM_COLUMNS }

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

5.12.1 Enumeration Type Documentation

5.12.1.1 anonymous enum

anonymous enum

Enumerator

COLUMN_SHORTCUT	
COLUMN_ACTION	
NUM_COLUMNS	

5.12.2 Function Documentation

5.12.2.1 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.

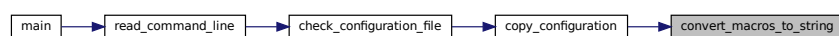
Must be NULL terminated

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

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

Referenced by `copy_configuration()`.

Here is the caller graph for this function:



5.12.2.2 convert_string_to_macros()

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

Convert the array of strings to macros.

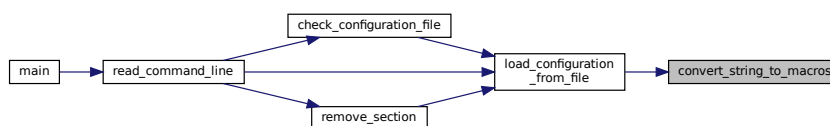
References [macro_t::action](#), [macros](#), [nr_of_macros](#), [remove_shortcuts\(\)](#), and [macro_t::shortcut](#).

Referenced by [load_configuration_from_file\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.12.2.3 get_shortcuts()

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

References [macros](#).

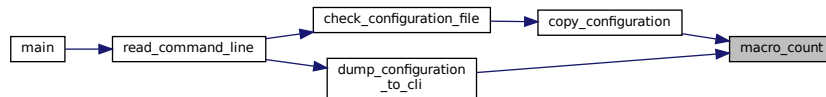
5.12.2.4 macro_count()

```
int macro_count ( )
```

References [nr_of_macros](#).

Referenced by [copy_configuration\(\)](#), and [dump_configuration_to_cli\(\)](#).

Here is the caller graph for this function:



5.12.2.5 remove_shortcuts()

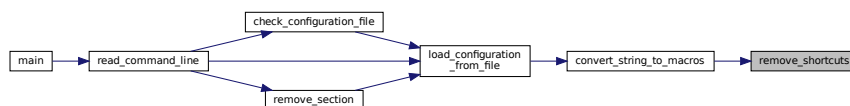
```
void remove_shortcuts (
    void )
```

Clean up all macros

References [macros](#).

Referenced by [convert_string_to_macros\(\)](#).

Here is the caller graph for this function:



5.12.3 Variable Documentation

5.12.3.1 macros

```
macro_t* macros = NULL
```

Referenced by [convert_macros_to_string\(\)](#), [convert_string_to_macros\(\)](#), [dump_configuration_to_cli\(\)](#), [get_shortcuts\(\)](#), and [remove_shortcuts\(\)](#).

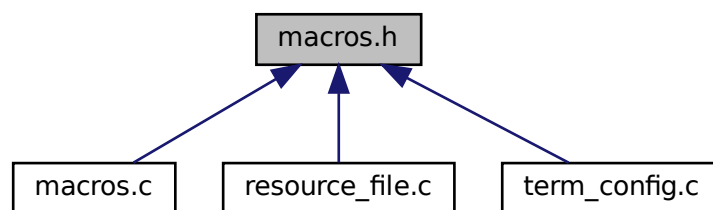
5.12.3.2 nr_of_macros

```
int nr_of_macros = 0
```

Referenced by [convert_macros_to_string\(\)](#), [convert_string_to_macros\(\)](#), and [macro_count\(\)](#).

5.13 macros.h File Reference

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



Classes

- struct [macro_t](#)
Define macro structure type.

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](#)

5.13.1 Function Documentation

5.13.1.1 `add_shortcuts()`

```
void add_shortcuts (
    void )
```

Remove shortcuts from `accel_group` and free memory.

5.13.1.2 `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.

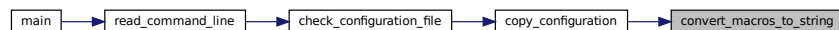
Must be NULL terminated

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

References [macro_t::action](#), [macros](#), [nr_of_macros](#), and [macro_t::shortcut](#).

Referenced by [copy_configuration\(\)](#).

Here is the caller graph for this function:



5.13.1.3 `convert_string_to_macros()`

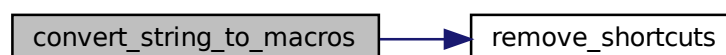
```
void convert_string_to_macros (
    char ** string_list,
    int size )
```

Convert the array of strings to macros.

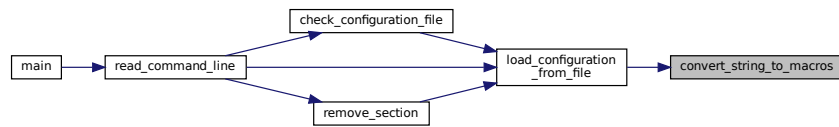
References [macro_t::action](#), [macros](#), [nr_of_macros](#), [remove_shortcuts\(\)](#), and [macro_t::shortcut](#).

Referenced by [load_configuration_from_file\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.13.1.4 `get_shortcuts()`

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

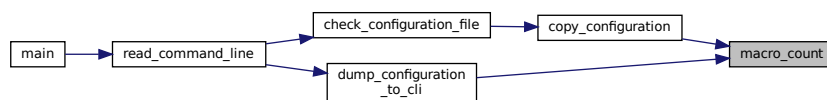
5.13.1.5 `macro_count()`

```
int macro_count ( )
```

References [nr_of_macros](#).

Referenced by [copy_configuration\(\)](#), and [dump_configuration_to_cli\(\)](#).

Here is the caller graph for this function:



5.13.1.6 `remove_shortcuts()`

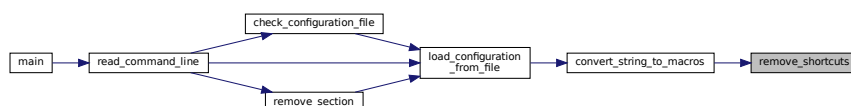
```
void remove_shortcuts (
    void )
```

Clean up all macros

References [macros](#).

Referenced by [convert_string_to_macros\(\)](#).

Here is the caller graph for this function:



5.13.2 Variable Documentation

5.13.2.1 macros

`macro_t* macros [extern]`

Referenced by [convert_macros_to_string\(\)](#), [convert_string_to_macros\(\)](#), [dump_configuration_to_cli\(\)](#), [get_shortcuts\(\)](#), and [remove_shortcuts\(\)](#).

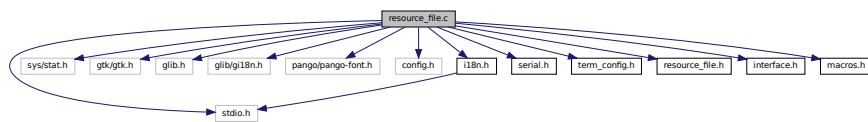
5.14 README.md File Reference

5.15 README.source File Reference

5.16 resource_file.c File Reference

```
#include <stdio.h>
#include <sys/stat.h>
#include <gtk/gtk.h>
#include <glib.h>
#include <glib/gi18n.h>
#include <pango/pango-font.h>
#include <config.h>
#include "i18n.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:



Macros

- `#define CONFIGURATION_FILENAME ".gtktermrc"`
Default configuration filename.
- `#define DEFAULT_SECTION "default"`

Enumerations

- enum {
[CONF_ITEM_PORT](#) , [CONF_ITEM_SPEED](#) , [CONF_ITEM_BITS](#) , [CONF_ITEM_STOPBITS](#) ,
[CONF_ITEM_PARITY](#) , [CONF_ITEM_FLOW_CONTROL](#) , [CONF_ITEM_WAIT_DELAY](#) , [CONF_ITEM_WAIT_CHAR](#)
 ,
[CONF_ITEM_RS485_RTS_TIME_BEFORE_TX](#) , [CONF_ITEM_RS485_RTS_TIME_AFTER_TX](#) , [CONF_ITEM_MACROS](#)
 , [CONF_ITEM_ECHO](#) ,
[CONF_ITEM_CRLF_AUTO](#) , [CONF_ITEM_DISABLE_PORT_LOCK](#) , [CONF_ITEM_FONT](#) , [CONF_ITEM_TERM_SHOW_CUR](#)
 ,
[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_FOREGR](#)
 , [CONF_ITEM_TERM_FOREGROUND_ALPHA](#) ,
[CONF_ITEM_TERM_BACKGROUND_RED](#) , [CONF_ITEM_TERM_BACKGROUND_GREEN](#) , [CONF_ITEM_TERM_BACKGR](#)
 , [CONF_ITEM_TERM_BACKGROUND_ALPHA](#) }

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

Functions

- void [config_file_init](#) (void)
- void [dump_configuration_to_cli](#) (char *section)
- void [save_configuration_to_file](#) (GKeyFile *config)
- int [load_configuration_from_file](#) (char *section)
- int [check_configuration_file](#) (void)
This checks if the configuration file exists.
- void [copy_configuration](#) (GKeyFile *configrc, const char *section)
Copy the active configuration into <section> of the Key file.
- int [remove_section](#) (char *section)
Remove a section from the GKeyFile. If it is the active section then switch back to default.
- void [hard_default_configuration](#) (void)
Create a new <default> configuration.
- void [validate_configuration](#) (void)
validate the active configuration
- void [set_color](#) (GdkRGBA *color, float R, float G, float B, float A)
Convert the colors RGB to internal color scheme.

Variables

- GFile * [config_file](#)
The key file.
- char [ConfigurationItem](#) [][32]

5.16.1 Macro Definition Documentation

5.16.1.1 CONFIGURATION_FILENAME

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

Default configuration filename.

5.16.1.2 DEFAULT_SECTION

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

5.16.2 Enumeration Type Documentation

5.16.2.1 anonymous enum

```
anonymous enum
```

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

it is an index to ConfigurationItem.

Enumerator

CONF_ITEM_PORT	
CONF_ITEM_SPEED	
CONF_ITEM_BITS	
CONF_ITEM_STOPBITS	
CONF_ITEM_PARITY	
CONF_ITEM_FLOW_CONTROL	
CONF_ITEM_WAIT_DELAY	
CONF_ITEM_WAIT_CHAR	
CONF_ITEM_RS485_RTS_TIME_BEFORE_TX	
CONF_ITEM_RS485_RTS_TIME_AFTER_TX	
CONF_ITEM_MACROS	
CONF_ITEM_ECHO	
CONF_ITEM_CRLF_AUTO	
CONF_ITEM_DISABLE_PORT_LOCK	
CONF_ITEM_FONT	
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	

5.16.3 Function Documentation

5.16.3.1 check_configuration_file()

```
int check_configuration_file (
    void )
```

This checks if the configuration file exists.

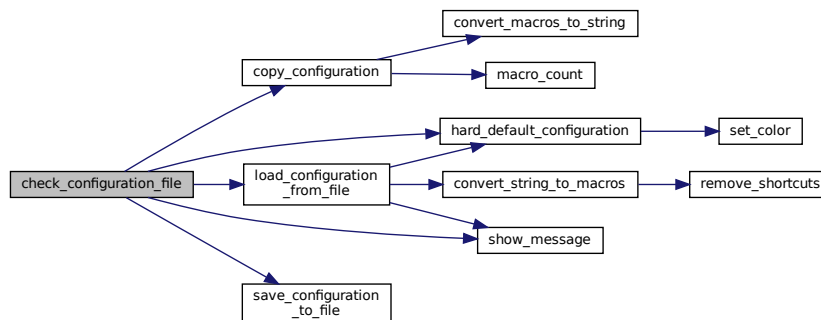
If not it creates a new [default] Put the new default in the key file

And save the config to file

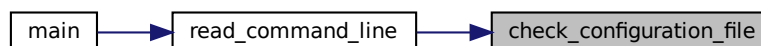
References [display_config_t::active_section](#), [config](#), [config_file](#), [copy_configuration\(\)](#), [DEFAULT_SECTION](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), [MSG_WRN](#), [save_configuration_to_file\(\)](#), [show_message\(\)](#), and [term_conf](#).

Referenced by [read_command_line\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.2 config_file_init()

```
void config_file_init (
    void )
```

References [config_file](#), and [CONFIGURATION_FILENAME](#).

Referenced by [main\(\)](#).

Here is the caller graph for this function:



5.16.3.3 copy_configuration()

```
void copy_configuration (
    GKeyFile * configrc,
    const char * section )
```

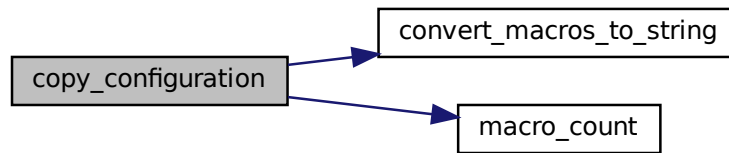
Copy the active configuration into <section> of the Key file.

Macros are an array of strings, so we have to convert it All macros ends up in the string_list

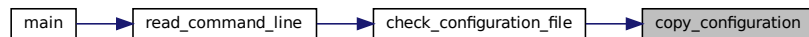
References [display_config_t::background_color](#), [port_config_t::bits](#), [display_config_t::char_queue](#), [display_config_t::columns](#), [CONF_ITEM_BITS](#), [CONF_ITEM_CRLF_AUTO](#), [CONF_ITEM_DISABLE_PORT_LOCK](#), [CONF_ITEM_ECHO](#), [CONF_ITEM_FLOW_CONTROL](#), [CONF_ITEM_FONT](#), [CONF_ITEM_MACROS](#), [CONF_ITEM_PARITY](#), [CONF_ITEM_PORT](#), [CONF_ITEM_RS485_RTS_TIME_AFTER_TX](#), [CONF_ITEM_RS485_RTS_TIME_BEFORE_TX](#), [CONF_ITEM_SPEED](#), [CONF_ITEM_STOPBITS](#), [CONF_ITEM_TERM_BACKGROUND_ALPHA](#), [CONF_ITEM_TERM_BACKGROUND_GREEN](#), [CONF_ITEM_TERM_BACKGROUND_RED](#), [CONF_ITEM_TERM_COLS](#), [CONF_ITEM_TERM_FOREGROUND_ALPHA](#), [CONF_ITEM_TERM_FOREGROUND_BLUE](#), [CONF_ITEM_TERM_FOREGROUND_RED](#), [CONF_ITEM_TERM_ROWS](#), [CONF_ITEM_TERM_SCROLLBACK](#), [CONF_ITEM_TERM_SHOW_CURSOR](#), [CONF_ITEM_TERM_VISUAL_BELL](#), [CONF_ITEM_WAIT_CHAR](#), [CONF_ITEM_WAIT_DELAY](#), [ConfigurationItem](#), [convert_macros_to_string\(\)](#), [display_config_t::crlfauto](#), [display_config_t::delay](#), [port_config_t::disable_port_lock](#), [display_config_t::echo](#), [port_config_t::flow_control](#), [display_config_t::font](#), [display_config_t::foreground_color](#), [macro_count\(\)](#), [port_config_t::parity](#), [port_config_t::port](#), [port_conf](#), [display_config_t::rows](#), [port_config_t::rs485_rts_time_after_transmit](#), [port_config_t::rs485_rts_time_before_transmit](#), [display_config_t::scrollback](#), [display_config_t::show_cursor](#), [port_config_t::speed](#), [port_config_t::stops](#), [term_conf](#), and [display_config_t::visual_bell](#).

Referenced by [check_configuration_file\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.4 dump_configuration_to_cli()

```
void dump_configuration_to_cli (
    char * section )
```

Print the serial port items

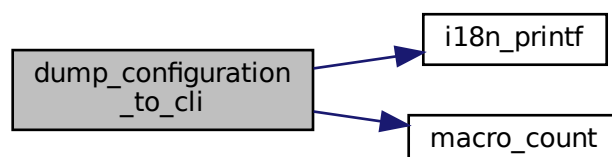
Print the terminal items

... and the macro's

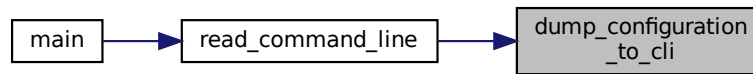
References [display_config_t::background_color](#), [port_config_t::bits](#), [display_config_t::block_cursor](#), [display_config_t::char_queue](#), [display_config_t::columns](#), [display_config_t::crlfauto](#), [display_config_t::delay](#), [port_config_t::disable_port_lock](#), [display_config_t::echo](#), [port_config_t::flow_control](#), [display_config_t::font](#), [display_config_t::foreground_color](#), [i18n_printf\(\)](#), [macro_count\(\)](#), [macros](#), [port_config_t::parity](#), [port_config_t::port](#), [port_conf](#), [display_config_t::rows](#), [port_config_t::rs485_rts_time_after_transmit](#), [port_config_t::rs485_rts_time_before_transmit](#), [display_config_t::scrollback](#), [display_config_t::show_cursor](#), [port_config_t::speed](#), [port_config_t::stops](#), [term_conf](#), [display_config_t::timestamp](#), and [display_config_t::visual_bell](#).

Referenced by [read_command_line\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.5 hard_default_configuration()

```
void hard_default_configuration (
    void )
```

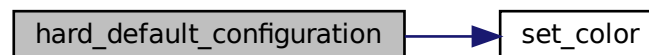
Create a new <default> configuration.

Store session, so we know in which active section we are

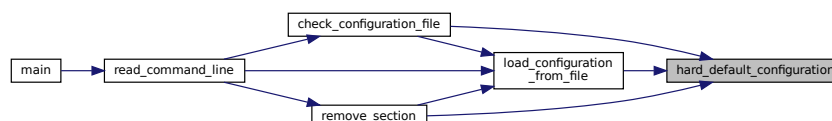
References [display_config_t::active_section](#), [display_config_t::background_color](#), [port_config_t::bits](#), [display_config_t::block_cursor](#), [display_config_t::char_queue](#), [display_config_t::columns](#), [display_config_t::crlfauto](#), [DEFAULT_BITS](#), [DEFAULT_CHAR](#), [DEFAULT_DELAY](#), [DEFAULT_DELAY_RS485](#), [DEFAULT_ECHO](#), [DEFAULT_FLOW](#), [DEFAULT_FONT](#), [DEFAULT_PARITY](#), [DEFAULT_PORT](#), [DEFAULT_SCROLLBACK](#), [DEFAULT_SECTION](#), [DEFAULT_SPEED](#), [DEFAULT_STOP](#), [display_config_t::delay](#), [port_config_t::disable_port_lock](#), [display_config_t::echo](#), [port_config_t::flow_control](#), [display_config_t::font](#), [display_config_t::foreground_color](#), [port_config_t::parity](#), [port_config_t::port](#), [port_conf](#), [display_config_t::rows](#), [port_config_t::rs485_rts_time_after_transmit](#), [port_config_t::rs485_rts_time_before_transmit](#), [display_config_t::scrollback](#), [set_color\(\)](#), [display_config_t::show_cursor](#), [port_config_t::speed](#), [port_config_t::stops](#), [term_conf](#), [display_config_t::timestamp](#), and [display_config_t::visual_bell](#).

Referenced by [check_configuration_file\(\)](#), [load_configuration_from_file\(\)](#), and [remove_section\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.6 load_configuration_from_file()

```
int load_configuration_from_file (
    char * section )
```

Load the key file Note: all sections are loaded into memory.

Check if the <section> exists in the key file.

First initialize with a default structure. Not really needed but good practice.

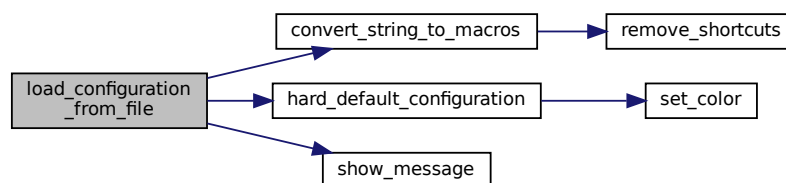
The Font is a Pango structure. This only can be added to a terminal So we have to convert it.

Convert the stringlist to macros. Existing shortcuts will be delete from convert_string_to_macros

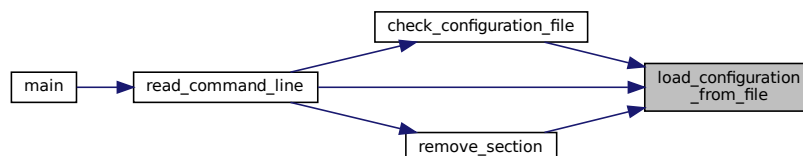
References [display_config_t::active_section](#), [display_config_t::background_color](#), [port_config_t::bits](#), [display_config_t::char_queue](#), [display_config_t::columns](#), [CONF_ITEM_BITS](#), [CONF_ITEM_CRLF_AUTO](#), [CONF_ITEM_DISABLE_PORT_LOCK](#), [CONF_ITEM_ECHO](#), [CONF_ITEM_FLOW_CONTROL](#), [CONF_ITEM_FONT](#), [CONF_ITEM_MACROS](#), [CONF_ITEM_PARITY](#), [CONF_ITEM_PORT](#), [CONF_ITEM_RS485_RTS_TIME_AFTER_TX](#), [CONF_ITEM_RS485_RTS_TIME_BEFORE_TX](#), [CONF_ITEM_SPEED](#), [CONF_ITEM_STOPBITS](#), [CONF_ITEM_TERM_BACKGROUND_ALPHA](#), [CONF_ITEM_TERM_BACKGROUND_GREEN](#), [CONF_ITEM_TERM_BACKGROUND_RED](#), [CONF_ITEM_TERM_COLS](#), [CONF_ITEM_TERM_FOREGROUND_ALPHA](#), [CONF_ITEM_TERM_FOREGROUND_BLUE](#), [CONF_ITEM_TERM_FOREGROUND_RED](#), [CONF_ITEM_TERM_ROWS](#), [CONF_ITEM_TERM_SCROLLBACK](#), [CONF_ITEM_TERM_SHOW_CURSOR](#), [CONF_ITEM_TERM_VISUAL_BELL](#), [CONF_ITEM_WAIT_CHAR](#), [CONF_ITEM_WAIT_DELAY](#), [config_file](#), [ConfigurationItem](#), [convert_string_to_macros\(\)](#), [display_config_t::crlfauto](#), [display_config_t::delay](#), [port_config_t::disable_port_lock](#), [display_config_t::echo](#), [port_config_t::flow_control](#), [display_config_t::font](#), [display_config_t::foreground_color](#), [hard_default_configuration\(\)](#), [MSG_ERR](#), [port_config_t::parity](#), [port_config_t::port](#), [port_conf](#), [display_config_t::rows](#), [port_config_t::rs485_rts_time_after_transmit](#), [port_config_t::rs485_rts_time_before_transmit](#), [display_config_t::scrollback](#), [display_config_t::show_cursor](#), [show_message\(\)](#), [port_config_t::speed](#), [port_config_t::stops](#), [term_conf](#), and [display_config_t::visual_bell](#).

Referenced by [check_configuration_file\(\)](#), [read_command_line\(\)](#), and [remove_section\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.7 remove_section()

```
int remove_section (
    char * section )
```

Remove a section from the GKeyFile If it is the active section then switch back to default.

If it is the default section then create a new 'default' default section Load the key file Note: all sections are loaded into memory. TODO: make it a own function if this works.....

If we remove the DEFAULT_SECTION then create a new one

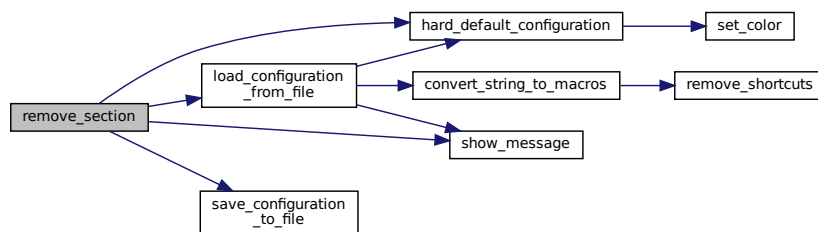
We remove the active session? Switch back to DEFAULT_SECTION

Remove the group from GKeyFile

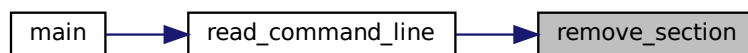
References [display_config_t::active_section](#), [config](#), [config_file](#), [DEFAULT_SECTION](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), [MSG_WRN](#), [save_configuration_to_file\(\)](#), [show_message\(\)](#), and [term_conf](#).

Referenced by [read_command_line\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



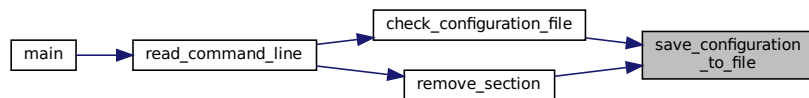
5.16.3.8 save_configuration_to_file()

```
void save_configuration_to_file (
    GKeyFile * config )
```

References [config](#), and [config_file](#).

Referenced by [check_configuration_file\(\)](#), and [remove_section\(\)](#).

Here is the caller graph for this function:



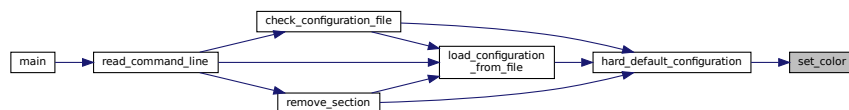
5.16.3.9 set_color()

```
void set_color (
    GdkRGBA * color,
    float R,
    float G,
    float B,
    float A )
```

Convert the colors RGB to internal color scheme.

Referenced by [hard_default_configuration\(\)](#).

Here is the caller graph for this function:



5.16.3.10 validate_configuration()

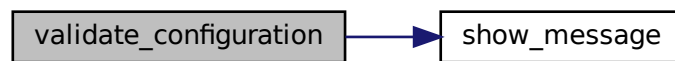
```
void validate_configuration (
    void )
```

validate the active configuration

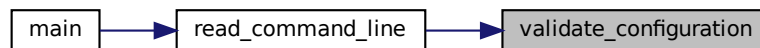
References [port_config_t::bits](#), [DEFAULT_BITS](#), [DEFAULT_DELAY](#), [DEFAULT_FONT](#), [DEFAULT_STOP](#), [display_config_t::delay](#), [display_config_t::font](#), [MSG_ERR](#), [port_conf](#), [show_message\(\)](#), [port_config_t::speed](#), [port_config_t::stops](#), and [term_conf](#).

Referenced by [read_command_line\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.4 Variable Documentation

5.16.4.1 config_file

```
GFile* config_file
```

The key file.

Referenced by [check_configuration_file\(\)](#), [config_file_init\(\)](#), [load_configuration_from_file\(\)](#), [remove_section\(\)](#), and [save_configuration_to_file\(\)](#).

5.16.4.2 ConfigurationItem

```
char ConfigurationItem[][32]
```

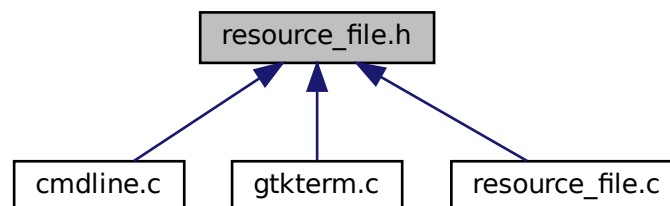
Initial value:

```
= {
    "port",
    "speed",
    "bits",
    "stopbits",
    "parity",
    "flow_control",
    "wait_delay",
    "wait_char",
    "rs485_rts_time_before_tx",
    "rs485_rts_time_after_tx",
    "macros",
    "echo",
    "crlfauto",
    "disable_port_lock",
    "term_font",
    "term_show_cursor",
    "term_rows",
    "term_columns",
    "term_scrollback",
    "term_visual_bell",
    "term_foreground_red",
    "term_foreground_green",
    "term_foreground_blue",
    "term_foreground_alpha",
    "term_background_red",
    "term_background_green",
    "term_background_blue",
    "term_background_alpha"
}
```

Referenced by [copy_configuration\(\)](#), and [load_configuration_from_file\(\)](#).

5.17 resource_file.h File Reference

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



Functions

- void [config_file_init](#) (void)
- void [save_configuration_to_file](#) (GKeyFile *)
- int [load_configuration_from_file](#) (char *)
- int [check_configuration_file](#) ()

- *This checks if the configuration file exists.*
- void [dump_configuration_to_cli](#) (char *)
- void [hard_default_configuration](#) (void)
 - *Create a new <default> configuration.*
- void [validate_configuration](#) (void)
 - *validate the active configuration*
- void [copy_configuration](#) (GKeyFile *, const char *)
 - *Copy the active configuration into <section> of the Key file.*
- int [remove_section](#) (char *)
 - *Remove a section from the GKeyFile If it is the active section then switch back to default.*
- void [set_color](#) (GdkRGBA *color, float, float, float, float)
 - *Convert the colors RGB to internal color scheme.*

Variables

- GFile * [config_file](#)
 - *The key file.*

5.17.1 Function Documentation

5.17.1.1 check_configuration_file()

```
int check_configuration_file (
    void )
```

This checks if the configuration file exists.

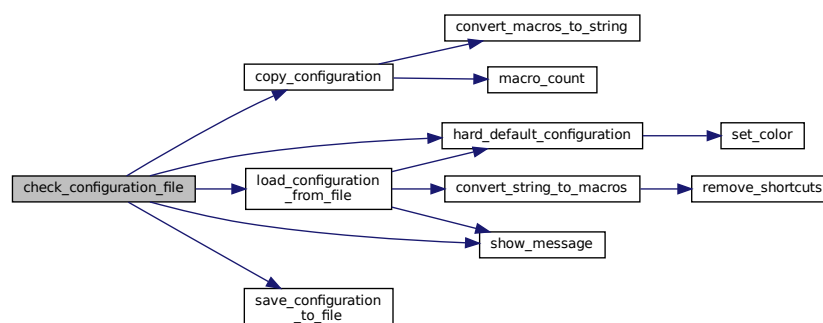
If not it creates a new [default] Put the new default in the key file

And save the config to file

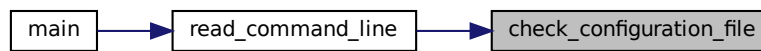
References [display_config_t::active_section](#), [config](#), [config_file](#), [copy_configuration\(\)](#), [DEFAULT_SECTION](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), [MSG_WRN](#), [save_configuration_to_file\(\)](#), [show_message\(\)](#), and [term_conf](#).

Referenced by [read_command_line\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.17.1.2 config_file_init()

```
void config_file_init (
    void )
```

References [config_file](#), and [CONFIGURATION_FILENAME](#).

Referenced by [main\(\)](#).

Here is the caller graph for this function:



5.17.1.3 copy_configuration()

```
void copy_configuration (
    GKeyFile * configrc,
    const char * section )
```

Copy the active configuration into <section> of the Key file.

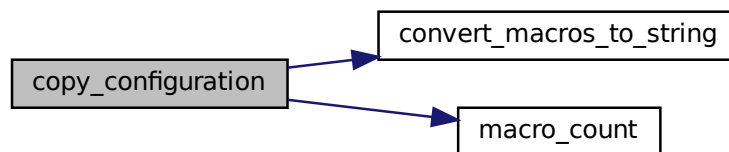
Macros are an array of strings, so we have to convert it All macros ends up in the string_list

References [display_config_t::background_color](#), [port_config_t::bits](#), [display_config_t::char_queue](#), [display_config_t::columns](#), [CONF_ITEM_BITS](#), [CONF_ITEM_CRLF_AUTO](#), [CONF_ITEM_DISABLE_PORT_LOCK](#), [CONF_ITEM_ECHO](#), [CONF_ITEM_FLOW_CONTROL](#), [CONF_ITEM_FONT](#), [CONF_ITEM_MACROS](#), [CONF_ITEM_PARITY](#), [CONF_ITEM_PORT](#), [CONF_ITEM_RS485_RTS_TIME_AFTER_TX](#), [CONF_ITEM_RS485_RTS_TIME_BEFORE_TX](#), [CONF_ITEM_SPEED](#), [CONF_ITEM_STOPBITS](#), [CONF_ITEM_TERM_BACKGROUND_ALPHA](#), [CONF_ITEM_TERM_BACKGROUND_GREEN](#), [CONF_ITEM_TERM_BACKGROUND_RED](#), [CONF_ITEM_TERM_COLS](#), [CONF_ITEM_TERM_FOREGROUND_ALPHA](#), [CONF_ITEM_TERM_FOREGROUND_BLUE](#), [CONF_ITEM_TERM_FOREGROUND](#)

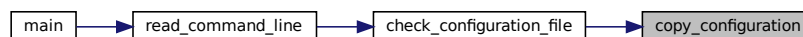
[CONF_ITEM_TERM_FOREGROUND_RED](#), [CONF_ITEM_TERM_ROWS](#), [CONF_ITEM_TERM_SCROLLBACK](#), [CONF_ITEM_TERM_SHOW_CURSOR](#), [CONF_ITEM_TERM_VISUAL_BELL](#), [CONF_ITEM_WAIT_CHAR](#), [CONF_ITEM_WAIT_DELAY](#), [ConfigurationItem](#), [convert_macros_to_string\(\)](#), [display_config_t::crlfauto](#), [display_config_t::delay](#), [port_config_t::disable_port_lock](#), [display_config_t::echo](#), [port_config_t::flow_control](#), [display_config_t::font](#), [display_config_t::foreground_color](#), [macro_count\(\)](#), [port_config_t::parity](#), [port_config_t::port](#), [port_conf](#), [display_config_t::rows](#), [port_config_t::rs485_rts_time_after_transmit](#), [port_config_t::rs485_rts_time_before_transmit](#), [display_config_t::scrollback](#), [display_config_t::show_cursor](#), [port_config_t::speed](#), [port_config_t::stops](#), [term_conf](#), and [display_config_t::visual_bell](#).

Referenced by [check_configuration_file\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.17.1.4 dump_configuration_to_cli()

```
void dump_configuration_to_cli (
    char * section )
```

Print the serial port items

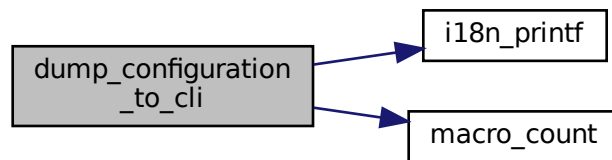
Print the terminal items

... and the macro's

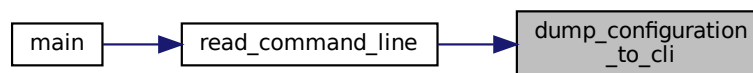
References [display_config_t::background_color](#), [port_config_t::bits](#), [display_config_t::block_cursor](#), [display_config_t::char_queue](#), [display_config_t::columns](#), [display_config_t::crlfauto](#), [display_config_t::delay](#), [port_config_t::disable_port_lock](#), [display_config_t::echo](#), [port_config_t::flow_control](#), [display_config_t::font](#), [display_config_t::foreground_color](#), [i18n_printf\(\)](#), [macro_count\(\)](#), [macros](#), [port_config_t::parity](#), [port_config_t::port](#), [port_conf](#), [display_config_t::rows](#), [port_config_t::rs485_rts_time_after_transmit](#), [port_config_t::rs485_rts_time_before_transmit](#), [display_config_t::scrollback](#), [display_config_t::show_cursor](#), [port_config_t::speed](#), [port_config_t::stops](#), [term_conf](#), [display_config_t::timestamp](#), and [display_config_t::visual_bell](#).

Referenced by [read_command_line\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.17.1.5 hard_default_configuration()

```
void hard_default_configuration (
    void )
```

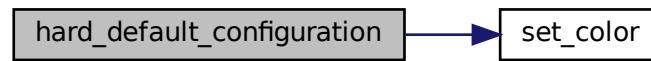
Create a new <default> configuration.

Store session, so we know in which active section we are

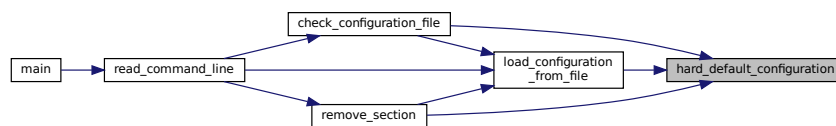
References [display_config_t::active_section](#), [display_config_t::background_color](#), [port_config_t::bits](#), [display_config_t::block_cursor](#), [display_config_t::char_queue](#), [display_config_t::columns](#), [display_config_t::crlfauto](#), [DEFAULT_BITS](#), [DEFAULT_CHAR](#), [DEFAULT_DELAY](#), [DEFAULT_DELAY_RS485](#), [DEFAULT_ECHO](#), [DEFAULT_FLOW](#), [DEFAULT_FONT](#), [DEFAULT_PARITY](#), [DEFAULT_PORT](#), [DEFAULT_SCROLLBACK](#), [DEFAULT_SECTION](#), [DEFAULT_SPEED](#), [DEFAULT_STOP](#), [display_config_t::delay](#), [port_config_t::disable_port_lock](#), [display_config_t::echo](#), [port_config_t::flow_control](#), [display_config_t::font](#), [display_config_t::foreground_color](#), [port_config_t::parity](#), [port_config_t::port](#), [port_conf](#), [display_config_t::rows](#), [port_config_t::rs485_rts_time_after_transmit](#), [port_config_t::rs485_rts_time_before_transmit](#), [display_config_t::scrollback](#), [set_color\(\)](#), [display_config_t::show_cursor](#), [port_config_t::speed](#), [port_config_t::stops](#), [term_conf](#), [display_config_t::timestamp](#), and [display_config_t::visual_bell](#).

Referenced by [check_configuration_file\(\)](#), [load_configuration_from_file\(\)](#), and [remove_section\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.17.1.6 load_configuration_from_file()

```
int load_configuration_from_file (
    char * section )
```

Load the key file Note: all sections are loaded into memory.

Check if the <section> exists in the key file.

First initialize with a default structure. Not really needed but good practice.

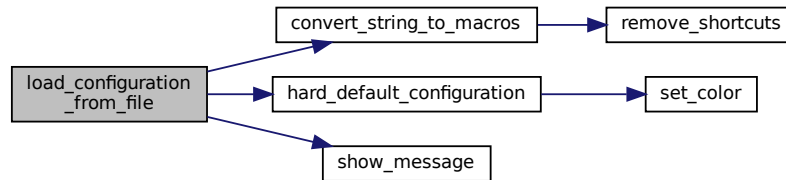
The Font is a Pango structure. This only can be added to a terminal So we have to convert it.

Convert the stringlist to macros. Existing shortcuts will be delete from convert_string_to_macros

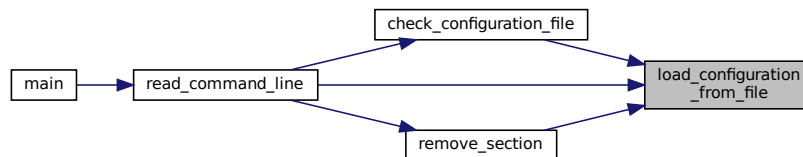
References [display_config_t::active_section](#), [display_config_t::background_color](#), [port_config_t::bits](#), [display_config_t::char_queue](#), [display_config_t::columns](#), [CONF_ITEM_BITS](#), [CONF_ITEM_CRLF_AUTO](#), [CONF_ITEM_DISABLE_PORT_LOCK](#), [CONF_ITEM_ECHO](#), [CONF_ITEM_FLOW_CONTROL](#), [CONF_ITEM_FONT](#), [CONF_ITEM_MACROS](#), [CONF_ITEM_PARITY](#), [CONF_ITEM_PORT](#), [CONF_ITEM_RS485_RTS_TIME_AFTER_TX](#), [CONF_ITEM_RS485_RTS_TIME_BEFORE_TX](#), [CONF_ITEM_SPEED](#), [CONF_ITEM_STOPBITS](#), [CONF_ITEM_TERM_BACKGROUND_ALPHA](#), [CONF_ITEM_TERM_BACKGROUND_GREEN](#), [CONF_ITEM_TERM_BACKGROUND_RED](#), [CONF_ITEM_TERM_COLS](#), [CONF_ITEM_TERM_FOREGROUND_ALPHA](#), [CONF_ITEM_TERM_FOREGROUND_BLUE](#), [CONF_ITEM_TERM_FOREGROUND_RED](#), [CONF_ITEM_TERM_ROWS](#), [CONF_ITEM_TERM_SCROLLBACK](#), [CONF_ITEM_TERM_SHOW_CURSOR](#), [CONF_ITEM_TERM_VISUAL_BELL](#), [CONF_ITEM_WAIT_CHAR](#), [CONF_ITEM_WAIT_DELAY](#), [config_file](#), [ConfigurationItem](#), [convert_string_to_macros\(\)](#), [display_config_t::crlfauto](#), [display_config_t::delay](#), [port_config_t::disable_port_lock](#), [display_config_t::echo](#), [port_config_t::flow_control](#), [display_config_t::font](#), [display_config_t::foreground_color](#), [hard_default_configuration\(\)](#), [MSG_ERR](#), [port_config_t::parity](#), [port_config_t::port](#), [port_conf](#), [display_config_t::rows](#), [port_config_t::rs485_rts_time_after_transmit](#), [port_config_t::rs485_rts_time_before_transmit](#), [display_config_t::scrollback](#), [display_config_t::show_cursor](#), [show_message\(\)](#), [port_config_t::speed](#), [port_config_t::stops](#), [term_conf](#), and [display_config_t::visual_bell](#).

Referenced by [check_configuration_file\(\)](#), [read_command_line\(\)](#), and [remove_section\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.17.1.7 remove_section()

```
int remove_section (
    char * section )
```

Remove a section from the GKeyFile If it is the active section then switch back to default.

If it is the default section then create a new 'default' default section Load the key file Note: all sections are loaded into memory. TODO: make it a own function if this works.....

If we remove the DEFAULT_SECTION then create a new one

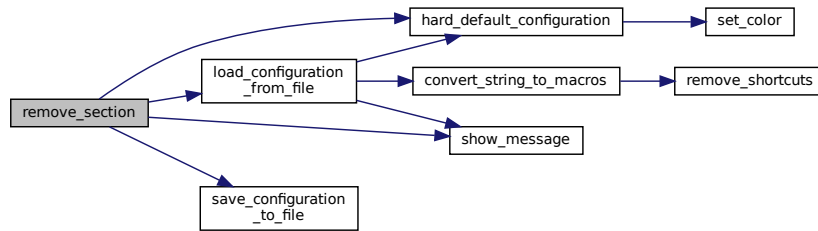
We remove the active session? Switch back to DEFAULT_SECTION

Remove the group from GKeyFile

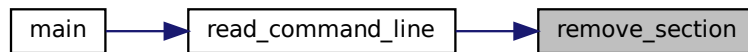
References [display_config_t::active_section](#), [config](#), [config_file](#), [DEFAULT_SECTION](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), [MSG_WRN](#), [save_configuration_to_file\(\)](#), [show_message\(\)](#), and [term_conf](#).

Referenced by [read_command_line\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



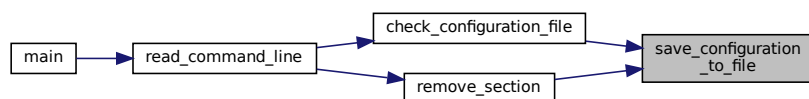
5.17.1.8 save_configuration_to_file()

```
void save_configuration_to_file (
    GKeyFile * config )
```

References [config](#), and [config_file](#).

Referenced by [check_configuration_file\(\)](#), and [remove_section\(\)](#).

Here is the caller graph for this function:



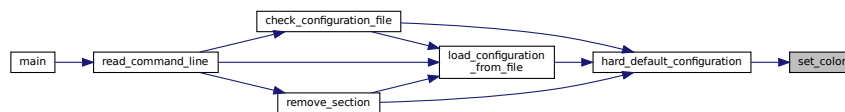
5.17.1.9 set_color()

```
void set_color (
    GdkRGBA * color,
    float R,
    float G,
    float B,
    float A )
```

Convert the colors RGB to internal color scheme.

Referenced by [hard_default_configuration\(\)](#).

Here is the caller graph for this function:



5.17.1.10 validate_configuration()

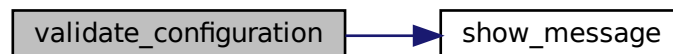
```
void validate_configuration (
    void )
```

validate the active configuration

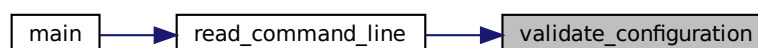
References [port_config_t::bits](#), [DEFAULT_BITS](#), [DEFAULT_DELAY](#), [DEFAULT_FONT](#), [DEFAULT_STOP](#), [display_config_t::delay](#), [display_config_t::font](#), [MSG_ERR](#), [port_conf](#), [show_message\(\)](#), [port_config_t::speed](#), [port_config_t::stops](#), and [term_conf](#).

Referenced by [read_command_line\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.17.2 Variable Documentation

5.17.2.1 config_file

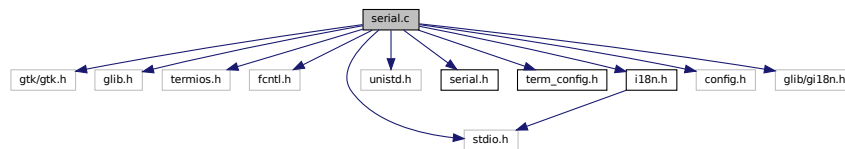
```
GFile* config_file [extern]
```

The key file.

Referenced by [check_configuration_file\(\)](#), [config_file_init\(\)](#), [load_configuration_from_file\(\)](#), [remove_section\(\)](#), and [save_configuration_to_file\(\)](#).

5.18 serial.c File Reference

```
#include <gtk/gtk.h>
#include <glib.h>
#include <termios.h>
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include "serial.h"
#include "term_config.h"
#include "i18n.h"
#include <config.h>
#include <glib/gi18n.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

5.18.1 Function Documentation

5.18.1.1 `get_port_string()`

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

References [port_config_t::bits](#), [port_config_t::parity](#), [port_config_t::port](#), [port_conf](#), [serial_port_fd](#), [port_config_t::speed](#), and [port_config_t::stops](#).

Referenced by [set_window_title\(\)](#).

Here is the caller graph for this function:



5.18.2 Variable Documentation

5.18.2.1 `port_conf`

```
port_config_t port_conf
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [get_port_string\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), [read_command_line\(\)](#), and [validate_configuration\(\)](#).

5.18.2.2 `serial_port_fd`

```
int serial_port_fd = -1
```

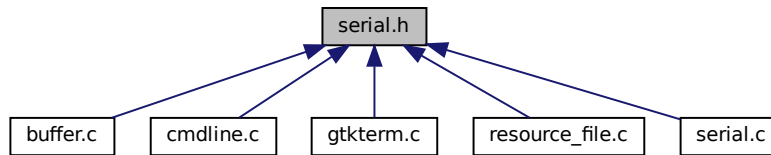
Referenced by [get_port_string\(\)](#).

5.18.2.3 `termios_save`

```
struct termios termios_save
```

5.19 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_SPEED](#) `115200`
- #define [DEFAULT_PARITY](#) `0`
- #define [DEFAULT_BITS](#) `8`
- #define [DEFAULT_STOP](#) `1`
- #define [DEFAULT_FLOW](#) `0`
- #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](#)

5.19.1 Macro Definition Documentation

5.19.1.1 DEFAULT_BITS

```
#define DEFAULT_BITS 8
```

5.19.1.2 DEFAULT_FLOW

```
#define DEFAULT_FLOW 0
```

5.19.1.3 DEFAULT_PARITY

```
#define DEFAULT_PARITY 0
```

5.19.1.4 DEFAULT_PORT

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

5.19.1.5 DEFAULT_SPEED

```
#define DEFAULT_SPEED 115200
```

5.19.1.6 DEFAULT_STOP

```
#define DEFAULT_STOP 1
```

5.19.1.7 LINE_FEED

```
#define LINE_FEED 0x0A
```

5.19.1.8 POLL_DELAY

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

5.19.1.9 RECEIVE_BUFFER

```
#define RECEIVE_BUFFER 8192
```

5.19.1.10 TRANSMIT_BUFFER

```
#define TRANSMIT_BUFFER 4096
```

5.19.2 Function Documentation

5.19.2.1 get_port_string()

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

References [port_config_t::bits](#), [port_config_t::parity](#), [port_config_t::port](#), [port_conf](#), [serial_port_fd](#), [port_config_t::speed](#), and [port_config_t::stops](#).

Referenced by [set_window_title\(\)](#).

Here is the caller graph for this function:



5.19.3 Variable Documentation

5.19.3.1 port_conf

```
port_config_t port_conf [extern]
```

Referenced by [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [get_port_string\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), [read_command_line\(\)](#), and [validate_configuration\(\)](#).

5.19.3.2 serial_port_fd

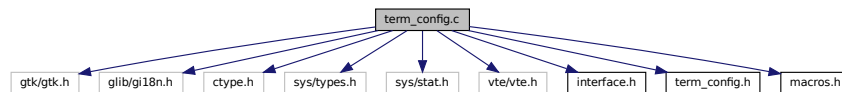
```
int serial_port_fd [extern]
```

Referenced by [get_port_string\(\)](#).

5.20 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](#)

5.20.1 Variable Documentation

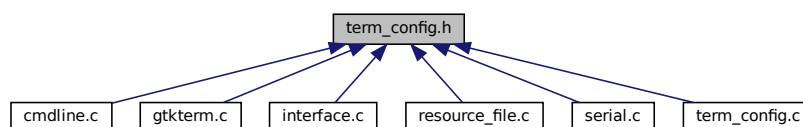
5.20.1.1 term_conf

```
display_config_t term_conf
```

Referenced by [check_configuration_file\(\)](#), [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), [read_command_line\(\)](#), [remove_section\(\)](#), and [validate_configuration\(\)](#).

5.21 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

Variables

- [display_config_t](#) `term_conf`

5.21.1 Macro Definition Documentation

5.21.1.1 DEFAULT_CHAR

```
#define DEFAULT_CHAR -1
```

5.21.1.2 DEFAULT_DELAY

```
#define DEFAULT_DELAY 0
```

5.21.1.3 DEFAULT_DELAY_RS485

```
#define DEFAULT_DELAY_RS485 30
```

5.21.1.4 DEFAULT_ECHO

```
#define DEFAULT_ECHO FALSE
```

5.21.1.5 DEFAULT_FONT

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

5.21.1.6 DEFAULT_SCROLLBACK

```
#define DEFAULT_SCROLLBACK 10000
```

5.21.2 Variable Documentation

5.21.2.1 term_conf

```
display_config_t term_conf [extern]
```

Referenced by [check_configuration_file\(\)](#), [copy_configuration\(\)](#), [dump_configuration_to_cli\(\)](#), [hard_default_configuration\(\)](#), [load_configuration_from_file\(\)](#), [read_command_line\(\)](#), [remove_section\(\)](#), and [validate_configuration\(\)](#).

Index

- action
 - macro_t, [16](#)
- active_section
 - display_config_t, [10](#)
- add_input
 - files.h, [32](#)
- add_shortcuts
 - macros.h, [47](#)
- ASCII_VIEW
 - interface.h, [42](#)
- background_color
 - display_config_t, [10](#)
- bits
 - port_config_t, [17](#)
- block_cursor
 - display_config_t, [10](#)
- buffer
 - GtkTermWindow, [13](#)
- buffer.c, [21](#)
 - clear_buffer, [22](#)
 - clear_func, [24](#)
 - create_buffer, [22](#)
 - delete_buffer, [22](#)
 - insert_timestamp, [22](#)
 - overlapped, [24](#)
 - put_chars, [23](#)
 - set_clear_func, [23](#)
 - set_display_func, [23](#)
 - timestamp_on, [25](#)
 - TIMESTAMP_SIZE, [22](#)
 - unset_clear_func, [23](#)
 - unset_display_func, [23](#)
 - virt_col_pos, [25](#)
 - write_buffer, [23](#)
 - write_buffer_with_func, [24](#)
 - write_func, [25](#)
- buffer.h, [25](#)
 - BUFFER_SIZE, [26](#)
 - clear_buffer, [26](#)
 - create_buffer, [26](#)
 - delete_buffer, [26](#)
 - put_chars, [27](#)
 - set_clear_func, [27](#)
 - set_display_func, [27](#)
 - unset_clear_func, [27](#)
 - unset_display_func, [27](#)
 - write_buffer, [27](#)
 - write_buffer_with_func, [28](#)
- BUFFER_SIZE
 - buffer.h, [26](#)
- char_queue
 - display_config_t, [10](#)
 - port_config_t, [17](#)
- check_configuration_file
 - resource_file.c, [53](#)
 - resource_file.h, [62](#)
- clear_buffer
 - buffer.c, [22](#)
 - buffer.h, [26](#)
- clear_func
 - buffer.c, [24](#)
- closure
 - macro_t, [16](#)
- cmdline.c, [28](#)
 - config, [30](#)
 - display_help, [29](#)
 - read_command_line, [29](#)
- cmdline.h, [31](#)
 - read_command_line, [31](#)
- COLUMN_ACTION
 - macros.c, [44](#)
- COLUMN_SHORTCUT
 - macros.c, [44](#)
- columns
 - display_config_t, [10](#)
- CONF_ITEM_BITS
 - resource_file.c, [52](#)
- CONF_ITEM_CRLF_AUTO
 - resource_file.c, [52](#)
- CONF_ITEM_DISABLE_PORT_LOCK
 - resource_file.c, [52](#)
- CONF_ITEM_ECHO
 - resource_file.c, [52](#)
- CONF_ITEM_FLOW_CONTROL
 - resource_file.c, [52](#)
- CONF_ITEM_FONT
 - resource_file.c, [52](#)
- CONF_ITEM_MACROS
 - resource_file.c, [52](#)
- CONF_ITEM_PARITY
 - resource_file.c, [52](#)
- CONF_ITEM_PORT
 - resource_file.c, [52](#)
- CONF_ITEM_RS485_RTS_TIME_AFTER_TX
 - resource_file.c, [52](#)
- CONF_ITEM_RS485_RTS_TIME_BEFORE_TX
 - resource_file.c, [52](#)
- CONF_ITEM_SPEED

- resource_file.c, [52](#)
- CONF_ITEM_STOPBITS
 - resource_file.c, [52](#)
- CONF_ITEM_TERM_BACKGROUND_ALPHA
 - resource_file.c, [52](#)
- CONF_ITEM_TERM_BACKGROUND_BLUE
 - resource_file.c, [52](#)
- CONF_ITEM_TERM_BACKGROUND_GREEN
 - resource_file.c, [52](#)
- CONF_ITEM_TERM_BACKGROUND_RED
 - resource_file.c, [52](#)
- CONF_ITEM_TERM_COLS
 - resource_file.c, [52](#)
- CONF_ITEM_TERM_FOREGROUND_ALPHA
 - resource_file.c, [52](#)
- CONF_ITEM_TERM_FOREGROUND_BLUE
 - resource_file.c, [52](#)
- CONF_ITEM_TERM_FOREGROUND_GREEN
 - resource_file.c, [52](#)
- CONF_ITEM_TERM_FOREGROUND_RED
 - resource_file.c, [52](#)
- CONF_ITEM_TERM_ROWS
 - resource_file.c, [52](#)
- CONF_ITEM_TERM_SCROLLBACK
 - resource_file.c, [52](#)
- CONF_ITEM_TERM_SHOW_CURSOR
 - resource_file.c, [52](#)
- CONF_ITEM_TERM_VISUAL_BELL
 - resource_file.c, [52](#)
- CONF_ITEM_WAIT_CHAR
 - resource_file.c, [52](#)
- CONF_ITEM_WAIT_DELAY
 - resource_file.c, [52](#)
- config
 - cmdline.c, [30](#)
 - interface.c, [41](#)
- config_file
 - resource_file.c, [60](#)
 - resource_file.h, [70](#)
- config_file_init
 - resource_file.c, [53](#)
 - resource_file.h, [63](#)
- CONFIGURATION_FILENAME
 - resource_file.c, [51](#)
- ConfigurationItem
 - resource_file.c, [60](#)
- convert_macros_to_string
 - macros.c, [44](#)
 - macros.h, [48](#)
- convert_string_to_macros
 - macros.c, [44](#)
 - macros.h, [48](#)
- copy_configuration
 - resource_file.c, [54](#)
 - resource_file.h, [63](#)
- create_buffer
 - buffer.c, [22](#)
 - buffer.h, [26](#)
- crlfauto
 - display_config_t, [11](#)
- DEFAULT_BITS
 - serial.h, [72](#)
- DEFAULT_CHAR
 - term_config.h, [76](#)
- DEFAULT_DELAY
 - term_config.h, [76](#)
- DEFAULT_DELAY_RS485
 - term_config.h, [76](#)
- DEFAULT_ECHO
 - term_config.h, [76](#)
- default_filename
 - files.c, [32](#)
 - files.h, [33](#)
- DEFAULT_FLOW
 - serial.h, [73](#)
- DEFAULT_FONT
 - term_config.h, [76](#)
- DEFAULT_PARITY
 - serial.h, [73](#)
- DEFAULT_PORT
 - serial.h, [73](#)
- DEFAULT_SCROLLBACK
 - term_config.h, [77](#)
- DEFAULT_SECTION
 - resource_file.c, [52](#)
- DEFAULT_SPEED
 - serial.h, [73](#)
- DEFAULT_STOP
 - serial.h, [73](#)
- delay
 - display_config_t, [11](#)
- delete_buffer
 - buffer.c, [22](#)
 - buffer.h, [26](#)
- disable_port_lock
 - port_config_t, [18](#)
- display
 - interface.h, [43](#)
- display_config_t, [9](#)
 - active_section, [10](#)
 - background_color, [10](#)
 - block_cursor, [10](#)
 - char_queue, [10](#)
 - columns, [10](#)
 - crlfauto, [11](#)
 - delay, [11](#)
 - echo, [11](#)
 - font, [11](#)
 - foreground_color, [11](#)
 - rows, [11](#)
 - scrollback, [12](#)
 - show_cursor, [12](#)
 - timestamp, [12](#)
 - visual_bell, [12](#)
- display_help
 - cmdline.c, [29](#)

- dump_configuration_to_cli
 - resource_file.c, [55](#)
 - resource_file.h, [64](#)
- echo
 - display_config_t, [11](#)
- files.c, [31](#)
 - default_filename, [32](#)
- files.h, [32](#)
 - add_input, [32](#)
 - default_filename, [33](#)
 - save_raw_file, [33](#)
 - send_raw_file, [33](#)
 - waiting_for_char, [33](#)
- flow_control
 - port_config_t, [18](#)
- font
 - display_config_t, [11](#)
- foreground_color
 - display_config_t, [11](#)
- fullscreen
 - GtkTermWindow, [13](#)
- get_port_string
 - serial.c, [70](#)
 - serial.h, [74](#)
- get_shortcuts
 - macros.c, [45](#)
 - macros.h, [49](#)
- GtkTerm
 - gtkterm.c, [34](#)
- gtkterm.c, [34](#)
 - GtkTerm, [34](#)
 - GtkTermClass, [34](#)
 - GtkTermWindowClass, [35](#)
 - main, [35](#)
 - set_window_title, [35](#)
- GtkTermClass
 - gtkterm.c, [34](#)
- GtkTermWindow, [13](#)
 - buffer, [13](#)
 - fullscreen, [13](#)
 - height, [14](#)
 - infobar, [14](#)
 - maximized, [14](#)
 - menubutton, [14](#)
 - message, [14](#)
 - parent_instance, [14](#)
 - scrolled_window, [14](#)
 - status, [14](#)
 - toolmenu, [15](#)
 - width, [15](#)
- GtkTermWindowClass
 - gtkterm.c, [35](#)
- hard_default_configuration
 - resource_file.c, [56](#)
 - resource_file.h, [65](#)
- height
 - GtkTermWindow, [14](#)
- HEXADECIMAL_VIEW
 - interface.h, [42](#)
- i18n.c, [36](#)
 - i18n_fprintf, [36](#)
 - i18n_perror, [37](#)
 - i18n_printf, [37](#)
 - strerror_utf8, [37](#)
- i18n.h, [38](#)
 - i18n_fprintf, [39](#)
 - I18N_H, [38](#)
 - i18n_perror, [39](#)
 - i18n_printf, [39](#)
 - strerror_utf8, [39](#)
- i18n_fprintf
 - i18n.c, [36](#)
 - i18n.h, [39](#)
- I18N_H
 - i18n.h, [38](#)
- i18n_perror
 - i18n.c, [37](#)
 - i18n.h, [39](#)
- i18n_printf
 - i18n.c, [37](#)
 - i18n.h, [39](#)
- infobar
 - GtkTermWindow, [14](#)
- insert_timestamp
 - buffer.c, [22](#)
- interface.c, [40](#)
 - config, [41](#)
 - show_message, [40](#)
 - timestamp_on, [41](#)
 - virt_col_pos, [41](#)
- interface.h, [41](#)
 - ASCII_VIEW, [42](#)
 - display, [43](#)
 - HEXADECIMAL_VIEW, [42](#)
 - MSG_ERR, [42](#)
 - MSG_WRN, [42](#)
 - show_message, [42](#)
 - Text, [43](#)
- LINE_FEED
 - serial.h, [73](#)
- load_configuration_from_file
 - resource_file.c, [56](#)
 - resource_file.h, [66](#)
- macro_count
 - macros.c, [45](#)
 - macros.h, [49](#)
- macro_t, [15](#)
 - action, [16](#)
 - closure, [16](#)
 - shortcut, [16](#)
- macros

- macros.c, 46
 - macros.h, 50
- macros.c, 43
 - COLUMN_ACTION, 44
 - COLUMN_SHORTCUT, 44
 - convert_macros_to_string, 44
 - convert_string_to_macros, 44
 - get_shortcuts, 45
 - macro_count, 45
 - macros, 46
 - nr_of_macros, 46
 - NUM_COLUMNS, 44
 - remove_shortcuts, 46
- macros.h, 47
 - add_shortcuts, 47
 - convert_macros_to_string, 48
 - convert_string_to_macros, 48
 - get_shortcuts, 49
 - macro_count, 49
 - macros, 50
 - remove_shortcuts, 49
- main
 - gtkterm.c, 35
- maximized
 - GtkTermWindow, 14
- menubutton
 - GtkTermWindow, 14
- message
 - GtkTermWindow, 14
- MSG_ERR
 - interface.h, 42
- MSG_WRN
 - interface.h, 42
- nr_of_macros
 - macros.c, 46
- NUM_COLUMNS
 - macros.c, 44
- overlapped
 - buffer.c, 24
- parent_instance
 - GtkTermWindow, 14
- parity
 - port_config_t, 18
- POLL_DELAY
 - serial.h, 73
- port
 - port_config_t, 18
- port_conf
 - serial.c, 71
 - serial.h, 74
- port_config_t, 17
 - bits, 17
 - char_queue, 17
 - disable_port_lock, 18
 - flow_control, 18
 - parity, 18
 - port, 18
 - rs485_rts_time_after_transmit, 18
 - rs485_rts_time_before_transmit, 18
 - speed, 19
 - stops, 19
- put_chars
 - buffer.c, 23
 - buffer.h, 27
- read_command_line
 - cmdline.c, 29
 - cmdline.h, 31
- README.md, 50
- README.source, 50
- RECEIVE_BUFFER
 - serial.h, 73
- remove_section
 - resource_file.c, 57
 - resource_file.h, 67
- remove_shortcuts
 - macros.c, 46
 - macros.h, 49
- resource_file.c, 50
 - check_configuration_file, 53
 - CONF_ITEM_BITS, 52
 - CONF_ITEM_CRLF_AUTO, 52
 - CONF_ITEM_DISABLE_PORT_LOCK, 52
 - CONF_ITEM_ECHO, 52
 - CONF_ITEM_FLOW_CONTROL, 52
 - CONF_ITEM_FONT, 52
 - CONF_ITEM_MACROS, 52
 - CONF_ITEM_PARITY, 52
 - CONF_ITEM_PORT, 52
 - CONF_ITEM_RS485_RTS_TIME_AFTER_TX, 52
 - CONF_ITEM_RS485_RTS_TIME_BEFORE_TX, 52
 - CONF_ITEM_SPEED, 52
 - CONF_ITEM_STOPBITS, 52
 - CONF_ITEM_TERM_BACKGROUND_ALPHA, 52
 - CONF_ITEM_TERM_BACKGROUND_BLUE, 52
 - CONF_ITEM_TERM_BACKGROUND_GREEN, 52
 - CONF_ITEM_TERM_BACKGROUND_RED, 52
 - CONF_ITEM_TERM_COLS, 52
 - CONF_ITEM_TERM_FOREGROUND_ALPHA, 52
 - CONF_ITEM_TERM_FOREGROUND_BLUE, 52
 - CONF_ITEM_TERM_FOREGROUND_GREEN, 52
 - CONF_ITEM_TERM_FOREGROUND_RED, 52
 - CONF_ITEM_TERM_ROWS, 52
 - CONF_ITEM_TERM_SCROLLBACK, 52
 - CONF_ITEM_TERM_SHOW_CURSOR, 52
 - CONF_ITEM_TERM_VISUAL_BELL, 52
 - CONF_ITEM_WAIT_CHAR, 52
 - CONF_ITEM_WAIT_DELAY, 52
 - config_file, 60
 - config_file_init, 53
 - CONFIGURATION_FILENAME, 51
 - ConfigurationItem, 60

- copy_configuration, [54](#)
- DEFAULT_SECTION, [52](#)
- dump_configuration_to_cli, [55](#)
- hard_default_configuration, [56](#)
- load_configuration_from_file, [56](#)
- remove_section, [57](#)
- save_configuration_to_file, [58](#)
- set_color, [59](#)
- validate_configuration, [59](#)
- resource_file.h, [61](#)
 - check_configuration_file, [62](#)
 - config_file, [70](#)
 - config_file_init, [63](#)
 - copy_configuration, [63](#)
 - dump_configuration_to_cli, [64](#)
 - hard_default_configuration, [65](#)
 - load_configuration_from_file, [66](#)
 - remove_section, [67](#)
 - save_configuration_to_file, [68](#)
 - set_color, [68](#)
 - validate_configuration, [69](#)
- rows
 - display_config_t, [11](#)
- rs485_rts_time_after_transmit
 - port_config_t, [18](#)
- rs485_rts_time_before_transmit
 - port_config_t, [18](#)
- save_configuration_to_file
 - resource_file.c, [58](#)
 - resource_file.h, [68](#)
- save_raw_file
 - files.h, [33](#)
- scrollback
 - display_config_t, [12](#)
- scrolled_window
 - GtkTermWindow, [14](#)
- send_raw_file
 - files.h, [33](#)
- serial.c, [70](#)
 - get_port_string, [70](#)
 - port_conf, [71](#)
 - serial_port_fd, [71](#)
 - termios_save, [71](#)
- serial.h, [72](#)
 - DEFAULT_BITS, [72](#)
 - DEFAULT_FLOW, [73](#)
 - DEFAULT_PARITY, [73](#)
 - DEFAULT_PORT, [73](#)
 - DEFAULT_SPEED, [73](#)
 - DEFAULT_STOP, [73](#)
 - get_port_string, [74](#)
 - LINE_FEED, [73](#)
 - POLL_DELAY, [73](#)
 - port_conf, [74](#)
 - RECEIVE_BUFFER, [73](#)
 - serial_port_fd, [74](#)
 - TRANSMIT_BUFFER, [74](#)
- serial_port_fd
 - serial.c, [71](#)
 - serial.h, [74](#)
- set_clear_func
 - buffer.c, [23](#)
 - buffer.h, [27](#)
- set_color
 - resource_file.c, [59](#)
 - resource_file.h, [68](#)
- set_display_func
 - buffer.c, [23](#)
 - buffer.h, [27](#)
- set_window_title
 - gtkterm.c, [35](#)
- shortcut
 - macro_t, [16](#)
- show_cursor
 - display_config_t, [12](#)
- show_message
 - interface.c, [40](#)
 - interface.h, [42](#)
- speed
 - port_config_t, [19](#)
- status
 - GtkTermWindow, [14](#)
- stops
 - port_config_t, [19](#)
- strerror_utf8
 - i18n.c, [37](#)
 - i18n.h, [39](#)
- term_conf
 - term_config.c, [75](#)
 - term_config.h, [77](#)
- term_config.c, [75](#)
 - term_conf, [75](#)
- term_config.h, [75](#)
 - DEFAULT_CHAR, [76](#)
 - DEFAULT_DELAY, [76](#)
 - DEFAULT_DELAY_RS485, [76](#)
 - DEFAULT_ECHO, [76](#)
 - DEFAULT_FONT, [76](#)
 - DEFAULT_SCROLLBACK, [77](#)
 - term_conf, [77](#)
- termios_save
 - serial.c, [71](#)
- Text
 - interface.h, [43](#)
- timestamp
 - display_config_t, [12](#)
- timestamp_on
 - buffer.c, [25](#)
 - interface.c, [41](#)
- TIMESTAMP_SIZE
 - buffer.c, [22](#)
- toolmenu
 - GtkTermWindow, [15](#)
- TRANSMIT_BUFFER
 - serial.h, [74](#)

- unset_clear_func
 - buffer.c, [23](#)
 - buffer.h, [27](#)
- unset_display_func
 - buffer.c, [23](#)
 - buffer.h, [27](#)
- validate_configuration
 - resource_file.c, [59](#)
 - resource_file.h, [69](#)
- virt_col_pos
 - buffer.c, [25](#)
 - interface.c, [41](#)
- visual_bell
 - display_config_t, [12](#)
- waiting_for_char
 - files.h, [33](#)
- width
 - GtkTermWindow, [15](#)
- write_buffer
 - buffer.c, [23](#)
 - buffer.h, [27](#)
- write_buffer_with_func
 - buffer.c, [24](#)
 - buffer.h, [28](#)
- write_func
 - buffer.c, [25](#)