GTKTerm

Generated by Doxygen 1.9.4

1 GTKTerm: The source code architecture 1

1.1 General description . . . 1

1.2 Objects . . . 1

1.2.1 GtkTerm . . . 2

1.2.1.1 Members . . . 2

1.2.1.2 Signals . . . 2

1.2.1.3 Main functions . . . 2

1.2.2 GtkTermWindow . . . 2

1.2.2.1 Members . . . 2

1.2.2.2 Signals . . . 2

1.2.2.3 Main functions . . . 2

1.2.3 GtkTermTerminal . . . 2

1.2.3.1 Members . . . 2

1.2.3.2 Signals . . . 2

1.2.3.3 Main functions . . . 2

1.2.4 GtkTermConfiguration . . . 2

1.2.4.1 Members . . . 2

1.2.4.2 Signals . . . 2

1.2.4.3 Main functions . . . 2

1.2.5 GtkTermSerialPort . . . 2

1.2.5.1 Members . . . 2

1.2.5.2 Signals . . . 2

1.2.5.3 Main functions . . . 2

1.3 Links . . . 2

2 Todo List 3

3 Class Index 5

3.1 Class List . . . 5

4 File Index 7

4.1 File List . . . 7

5 Class Documentation 9

5.1 _GtkTerm Struct Reference . . . 9

5.1.1 Detailed Description . . . 10

5.1.2 Member Data Documentation . . . 10

5.1.2.1 action_group . . . 10

5.1.2.2 config . . . 10

5.1.2.3 g_config_group . . . 10

5.1.2.4 g_port_group . . . 11

5.1.2.5 g_term_group . . . 11

5.1.2.6 parent_instance . . . 11

5.1.2.7 section . . . 11

5.2 _GtkTermConfiguration Struct Reference 11

5.2.1 Member Data Documentation 12

5.2.1.1 parent_instance 12

5.3 _GtkTermConfigurationClass Struct Reference 12

5.3.1 Member Data Documentation 12

5.3.1.1 parent_class 12

5.4 _GtkTermSerialPort Struct Reference 13

5.4.1 Member Data Documentation 13

5.4.1.1 parent_instance 13

5.5 _GtkTermSerialPortClass Struct Reference 13

5.5.1 Member Data Documentation 14

5.5.1.1 parent_class 14

5.6 _GtkTermTerminal Struct Reference 14

5.6.1 Member Data Documentation 14

5.6.1.1 vte_object 15

5.7 _GtkTermTerminalClass Struct Reference 15

5.7.1 Member Data Documentation 15

5.7.1.1 vte_class 15

5.8 _GtkTermWindow Struct Reference 16

5.8.1 Detailed Description 17

5.8.2 Member Data Documentation 17

5.8.2.1 action_group 17

5.8.2.2 fullscreen 17

5.8.2.3 height 17

5.8.2.4 infobar 17

5.8.2.5 maximized 18

5.8.2.6 menubutton 18

5.8.2.7 message 18

5.8.2.8 parent_instance 18

5.8.2.9 scrolled_window 18

5.8.2.10 search_bar 18

5.8.2.11 status_config 19

5.8.2.12 status_config_message 19

5.8.2.13 status_message 19

5.8.2.14 status_serial_signal 19

5.8.2.15 statusbox 19

5.8.2.16 terminal_window 19

5.8.2.17 toolmenu 20

5.8.2.18 width 20

5.9 GtkTermConfigurationPrivate Struct Reference 20

5.9.1 Member Data Documentation 20

5.9.1.1 config_file 21

5.9.1.2 key_file 21
5.10 GtkTermSerialPortPrivate Struct Reference 21
5.10.1 Member Data Documentation 22
5.10.1.1 port_conf 22
5.10.1.2 serial_port_fd 22
5.10.1.3 termios_save 22
5.11 GtkTermTerminalPrivate Struct Reference 23
5.11.1 Member Data Documentation 23
5.11.1.1 app 24
5.11.1.2 macros 24
5.11.1.3 main_window 24
5.11.1.4 port_conf 24
5.11.1.5 section 24
5.11.1.6 serial_port 24
5.11.1.7 term_conf 25
5.11.1.8 view_mode 25
5.12 macro_t Struct Reference 25
5.12.1 Detailed Description 26
5.12.2 Member Data Documentation 26
5.12.2.1 action 26
5.12.2.2 closure 26
5.12.2.3 shortcut 26
5.13 port_config_t Struct Reference 27
5.13.1 Member Data Documentation 27
5.13.1.1 baudrate 27
5.13.1.2 bits 28
5.13.1.3 disable_port_lock 28
5.13.1.4 flow_control 28
5.13.1.5 parity 28
5.13.1.6 port 28
5.13.1.7 rs485_rts_time_after_transmit 28
5.13.1.8 rs485_rts_time_before_transmit 28
5.13.1.9 stopbits 29
5.14 term_config_t Struct Reference 29
5.14.1 Member Data Documentation 30
5.14.1.1 background_color 30
5.14.1.2 block_cursor 30
5.14.1.3 char_queue 30
5.14.1.4 columns 30
5.14.1.5 crlfauto 30
5.14.1.6 delay 30
5.14.1.7 echo 30

5.14.1.8 font 31

5.14.1.9 foreground_color 31

5.14.1.10 rows 31

5.14.1.11 scrollback 31

5.14.1.12 show_cursor 31

5.14.1.13 timestamp 31

5.14.1.14 visual_bell 31

6 File Documentation **33**

6.1 README_source.md File Reference 33

6.2 buffer.c File Reference 33

6.2.1 Macro Definition Documentation 34

6.2.1.1 MAX_SECTION_LENGTH 34

6.2.1.2 TIMESTAMP_SIZE 34

6.2.2 Function Documentation 34

6.2.2.1 clear_buffer() 34

6.2.2.2 create_buffer() 35

6.2.2.3 delete_buffer() 35

6.2.2.4 insert_timestamp() 35

6.2.2.5 put_chars() 35

6.2.2.6 set_clear_func() 35

6.2.2.7 set_display_func() 35

6.2.2.8 unset_clear_func() 36

6.2.2.9 unset_display_func() 36

6.2.2.10 write_buffer() 36

6.2.2.11 write_buffer_with_func() 36

6.2.3 Variable Documentation 37

6.2.3.1 clear_func 37

6.2.3.2 overlapped 37

6.2.3.3 timestamp_on 37

6.2.3.4 virt_col_pos 37

6.2.3.5 write_func 37

6.3 buffer.h File Reference 38

6.3.1 Macro Definition Documentation 38

6.3.1.1 BUFFER_SIZE 38

6.3.2 Function Documentation 38

6.3.2.1 clear_buffer() 39

6.3.2.2 create_buffer() 39

6.3.2.3 delete_buffer() 39

6.3.2.4 put_chars() 39

6.3.2.5 set_clear_func() 39

6.3.2.6 set_display_func() 39

6.3.2.7 unset_clear_func() 40

6.3.2.8 unset_display_func() 40

6.3.2.9 write_buffer() 40

6.3.2.10 write_buffer_with_func() 40

6.4 buffer.h 41

6.5 cmdline.c File Reference 41

6.5.1 Function Documentation 42

6.5.1.1 gtkterm_add_cmdline_options() 42

6.6 cmdline.h File Reference 42

6.6.1 Function Documentation 42

6.6.1.1 gtkterm_add_cmdline_options() 42

6.6.2 Variable Documentation 43

6.6.2.1 g_term_group 43

6.7 cmdline.h 43

6.8 defaults.h File Reference 43

6.8.1 Macro Definition Documentation 44

6.8.1.1 BUFFER_LENGTH 44

6.8.1.2 DEFAULT_BAUDRATE 44

6.8.1.3 DEFAULT_BITS 44

6.8.1.4 DEFAULT_CHAR 45

6.8.1.5 DEFAULT_DELAY 45

6.8.1.6 DEFAULT_DELAY_RS485 45

6.8.1.7 DEFAULT_ECHO 45

6.8.1.8 DEFAULT_FLOW 45

6.8.1.9 DEFAULT_FONT 45

6.8.1.10 DEFAULT_PARITY 45

6.8.1.11 DEFAULT_PORT 46

6.8.1.12 DEFAULT_SCROLLBACK 46

6.8.1.13 DEFAULT_STOPBITS 46

6.8.1.14 DEFAULT_VISUAL_BELL 46

6.8.1.15 LINE_FEED 46

6.8.1.16 MAX_SECTION_LENGTH 46

6.8.1.17 POLL_DELAY 46

6.8.1.18 RECEIVE_BUFFER 47

6.8.1.19 TRANSMIT_BUFFER 47

6.9 defaults.h 47

6.10 files.c File Reference 47

6.10.1 Variable Documentation 48

6.10.1.1 default_filename 48

6.11 files.h File Reference 48

6.11.1 Function Documentation 48

6.11.1.1 add_input() 48

6.11.1.2 save_raw_file() 48

6.11.1.3 send_raw_file() 49

6.11.2 Variable Documentation 49

6.11.2.1 default_filename 49

6.11.2.2 waiting_for_char 49

6.12 files.h 49

6.13 gtkterm.c File Reference 50

6.13.1 Function Documentation 50

6.13.1.1 main() 50

6.13.2 Variable Documentation 50

6.13.2.1 gtkterm_signals 51

6.14 gtkterm.h File Reference 51

6.14.1 Macro Definition Documentation 52

6.14.1.1 GTKTERM_TYPE_APP 52

6.14.2 Typedef Documentation 52

6.14.2.1 GtkTerm 52

6.14.3 Enumeration Type Documentation 52

6.14.3.1 anonymous enum 52

6.14.4 Variable Documentation 53

6.14.4.1 gtkterm_signals 53

6.15 gtkterm.h 53

6.16 gtkterm_window.c File Reference 54

6.16.1 Function Documentation 54

6.16.1.1 create_window() 55

6.16.1.2 set_window_title() 55

6.17 gtkterm_window.h File Reference 55

6.17.1 Macro Definition Documentation 56

6.17.1.1 GTKTERM_TYPE_GTKTERM_WINDOW 56

6.17.2 Typedef Documentation 56

6.17.2.1 GtkTermWindow 56

6.17.3 Function Documentation 56

6.17.3.1 create_window() 57

6.18 gtkterm_window.h 57

6.19 interface.c File Reference 57

6.19.1 Function Documentation 58

6.19.1.1 show_message() 58

6.19.2 Variable Documentation 58

6.19.2.1 timestamp_on 59

6.19.2.2 virt_col_pos 59

6.20 interface.h File Reference 59

6.20.1 Macro Definition Documentation 59

6.20.1.1 ASCII_VIEW 60

6.20.1.2 HEXADECIMAL_VIEW 60
6.20.1.3 MSG_ERR 60
6.20.1.4 MSG_WRN 60
6.20.2 Function Documentation 60
6.20.2.1 show_message() 60
6.20.3 Variable Documentation 61
6.20.3.1 display 61
6.20.3.2 Text 61
6.21 interface.h 61
6.22 macros.c File Reference 61
6.22.1 Enumeration Type Documentation 62
6.22.1.1 anonymous enum 62
6.22.2 Function Documentation 63
6.22.2.1 convert_macros_to_string() 63
6.22.2.2 convert_string_to_macros() 63
6.22.2.3 get_shortcuts() 63
6.22.2.4 macro_count() 64
6.22.2.5 remove_shortcuts() 64
6.22.3 Variable Documentation 64
6.22.3.1 macros 64
6.22.3.2 nr_of_macros 64
6.23 macros.h File Reference 65
6.23.1 Function Documentation 65
6.23.1.1 add_shortcuts() 65
6.23.1.2 convert_macros_to_string() 66
6.23.1.3 convert_string_to_macros() 66
6.23.1.4 get_shortcuts() 66
6.23.1.5 macro_count() 66
6.23.1.6 remove_shortcuts() 67
6.23.2 Variable Documentation 67
6.23.2.1 macros 67
6.24 macros.h 67
6.25 resource_file.c File Reference 68
6.25.1 Macro Definition Documentation 69
6.25.1.1 BUFFER_LENGTH 69
6.25.1.2 CONFIGURATION_FILENAME 69
6.25.2 Function Documentation 69
6.25.2.1 check_keyfile() 69
6.25.2.2 gtkterm_configuration_default_configuration() 70
6.25.2.3 gtkterm_configuration_validate() 70
6.25.2.4 on_set_config_options() 71
6.25.3 Variable Documentation 71

6.25.3.1 GtkTermConfigurationItems 72
6.26 resource_file.h File Reference 72
6.26.1 Macro Definition Documentation 73
6.26.1.1 CONF_ITEM_LENGTH 73
6.26.1.2 DEFAULT_SECTION 73
6.26.1.3 GTKTERM_TYPE_CONFIGURATION 74
6.26.2 Typedef Documentation 74
6.26.2.1 GtkTermConfiguration 74
6.26.3 Enumeration Type Documentation 74
6.26.3.1 anonymous enum 74
6.26.4 Function Documentation 75
6.26.4.1 gtkterm_configuration_new() 75
6.26.4.2 on_set_config_options() 75
6.26.5 Variable Documentation 76
6.26.5.1 GtkTermConfigurationItems 76
6.27 resource_file.h 76
6.28 serial.c File Reference 77
6.28.1 Enumeration Type Documentation 78
6.28.1.1 anonymous enum 78
6.28.2 Function Documentation 78
6.28.2.1 gtkterm_serial_port_get_string() 78
6.28.2.2 gtkterm_serial_port_new() 78
6.28.2.3 gtkterm_serial_port_status() 78
6.29 serial.h File Reference 79
6.29.1 Macro Definition Documentation 79
6.29.1.1 GTKTERM_TYPE_SERIAL_PORT 79
6.29.2 Typedef Documentation 79
6.29.2.1 GtkTermSerialPort 80
6.29.3 Function Documentation 80
6.29.3.1 gtkterm_serial_port_get_string() 80
6.29.3.2 gtkterm_serial_port_new() 80
6.29.3.3 gtkterm_serial_port_status() 80
6.30 serial.h 81
6.31 terminal.c File Reference 81
6.31.1 Enumeration Type Documentation 82
6.31.1.1 anonymous enum 82
6.31.2 Function Documentation 82
6.31.2.1 gtkterm_terminal_new() 83
6.32 terminal.h File Reference 83
6.32.1 Macro Definition Documentation 84
6.32.1.1 GTKTERM_TYPE_TERMINAL 84
6.32.2 Function Documentation 84

6.32.2.1 gtkterm_terminal_new() 84
6.33 terminal.h 85

Index **87**

Chapter 1

GTKTerm: The source code architecture

This file describes the architecture of GTKTerm. GtkTerm has several objects and uses signals to communicate between these objects.

One of the subgoals is not to use any global variables but exchange data by the use of signals. For that only the array of signals is a global variable.

Use of GTKTerm/GtkTerm/gtkterm naming schema: In this document several ways of Upper/Lowercase combinations of GTKTerm is used:

- GTKTerm: The name of the application
- GtkTerm: The first part of the name of the object in the source code. For example: GtkTermWindow.
- gtk_term: The first part of the function of an object in the source code. For example: gtkterm_window_init

1.1 General description

GTKTerm is build with the GTK4 framework. It uses Gobjects and communicates (mostly) through signals.

GTKTerm is the main application object. It is a holder for the keyfile. The commandline interfaces uses the application object framework to handle all commandline options. The options are connected to the relevant GObjects by signals. Almost all objects have a 'public' and 'private' part. However the 'public' part is not globally known (except for GtkTerm application object).

The core of the application is the terminal. This is a VTE object and handles all communication to and from the serial port. The terminal window holds the configuration of the terminal window and the serial ports. The configuraton is copied from the GtkTerm application which holds the keyfile. It is copied back to the keyfile if it is saved. For now the GtkTerm application has just one terminal window. The architecture of GTKTerm is able to support multiple terminal windows in future releases.

1.2 Objects

This part lists an overview of all objects used in GTKTerm. For details about implementation please use the GTKTERM.pdf which is a Doxygen generated overview of the GTKTerm source code.

1.2.1 GtkTerm

GtkTerm is the main GtkApplication object for GTKTerm. It starts the gtkterm_window and handles the cmdline interface (CLI). Options given at the CLI are directly stored into the in memory keyfile. This in memory keyfile is the base for the configuration of the terminal windows. Getting configuration for the terminal window is done by signals for the [section] needed.

1.2.1.1 Members

1.2.1.2 Signals

1.2.1.3 Main functions

1.2.2 GtkTermWindow

1.2.2.1 Members

1.2.2.2 Signals

1.2.2.3 Main functions

1.2.3 GtkTermTerminal

1.2.3.1 Members

1.2.3.2 Signals

1.2.3.3 Main functions

1.2.4 GtkTermConfiguration

1.2.4.1 Members

1.2.4.2 Signals

1.2.4.3 Main functions

1.2.5 GtkTermSerialPort

1.2.5.1 Members

1.2.5.2 Signals

1.2.5.3 Main functions

1.3 Links

For the migration to gtk4 several links were used:

- https://docs.gtk.org/gobject/tutorial.html
- https://docs.gtk.org/glib/
- https://toshiocp.github.io/Gtk4-tutorial/index.html
- https://c-gtk.org/gapplication-part-i

Also special thanks to Jens Georg. Sellerie (an earlier fork of GTKTerm) was used as inspiration to solve some problems.

Chapter 2

Todo List

Member GtkTermTerminalPrivate::macros

convert macros -> object

Chapter 3

Class Index

3.1 Class List

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

_GtkTerm
The main GtkTerm application class . . . 9
_GtkTermConfiguration . . . 11
_GtkTermConfigurationClass . . . 12
_GtkTermSerialPort . . . 13
_GtkTermSerialPortClass . . . 13
_GtkTermTerminal . . . 14
_GtkTermTerminalClass . . . 15
_GtkTermWindow
The main GtkTermWindow class . . . 16
GtkTermConfigurationPrivate . . . 20
GtkTermSerialPortPrivate . . . 21
GtkTermTerminalPrivate . . . 23
macro_t
Todo: Migrate to GObject . . . 25
port_config_t . . . 27
term_config_t . . . 29

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

buffer.c 33
buffer.h 38
cmdline.c 41
cmdline.h 42
defaults.h 43
files.c 47
files.h 48
gtkterm.c 50
gtkterm.h 51
gtkterm_window.c 54
gtkterm_window.h 55
interface.c 57
interface.h 59
macros.c 61
macros.h 65
resource_file.c 68
resource_file.h 72
serial.c 77
serial.h 79
terminal.c 81
terminal.h 83

Chapter 5

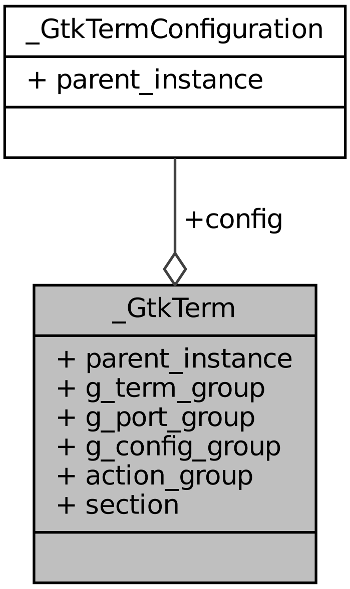
Class Documentation

5.1 _GtkTerm Struct Reference

The main GtkTerm application class.

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

Collaboration diagram for _GtkTerm:



Public Attributes

- GtkApplication parent_instance
- GOptionGroup ∗ g_term_group
- GOptionGroup ∗ g_port_group
- GOptionGroup ∗ g_config_group
- GActionGroup ∗ action_group

  App action group.
- GtkTermConfiguration ∗ config

  The Key file with the configurations.
- char ∗ section

  The section provided from the cli.

5.1.1 Detailed Description

The main GtkTerm application class.

All application specific variables are defined here.

5.1.2 Member Data Documentation

5.1.2.1 action_group

```
GActionGroup* _GtkTerm::action_group
```

App action group.

5.1.2.2 config

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

The Key file with the configurations.

5.1.2.3 g_config_group

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

Referenced by gtkterm_add_cmdline_options().

5.1.2.4 g_port_group

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

Referenced by gtkterm_add_cmdline_options().

5.1.2.5 g_term_group

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

Referenced by gtkterm_add_cmdline_options().

5.1.2.6 parent_instance

```
GtkApplication _GtkTerm::parent_instance
```

5.1.2.7 section

```
char* _GtkTerm::section
```

The section provided from the cli.

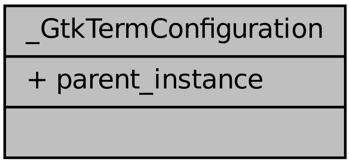
Terminals have their own section pointer

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

- gtkterm.h

5.2 _GtkTermConfiguration Struct Reference

Collaboration diagram for _GtkTermConfiguration:



Public Attributes

- GObject parent_instance

5.2.1 Member Data Documentation

5.2.1.1 parent_instance

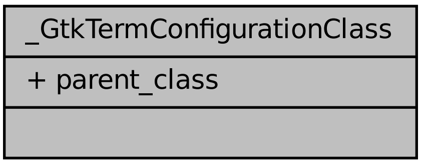
```
GObject _GtkTermConfiguration::parent_instance
```

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

- resource_file.c

5.3 _GtkTermConfigurationClass Struct Reference

Collaboration diagram for _GtkTermConfigurationClass:



Public Attributes

- GObjectClass parent_class

5.3.1 Member Data Documentation

5.3.1.1 parent_class

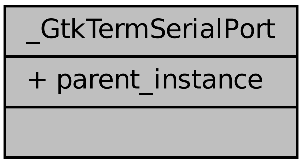
```
GObjectClass _GtkTermConfigurationClass::parent_class
```

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

- resource_file.c

5.4 _GtkTermSerialPort Struct Reference

Collaboration diagram for _GtkTermSerialPort:



Public Attributes

- GObject parent_instance

5.4.1 Member Data Documentation

5.4.1.1 parent_instance

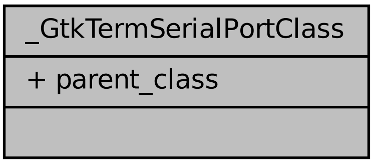
```
GObject _GtkTermSerialPort::parent_instance
```

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

- serial.c

5.5 _GtkTermSerialPortClass Struct Reference

Collaboration diagram for _GtkTermSerialPortClass:



Public Attributes

- GObjectClass parent_class

5.5.1 Member Data Documentation

5.5.1.1 parent_class

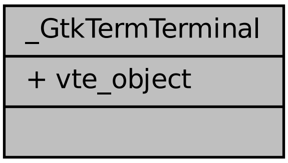
```
GObjectClass _GtkTermSerialPortClass::parent_class
```

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

- serial.c

5.6 _GtkTermTerminal Struct Reference

Collaboration diagram for _GtkTermTerminal:



Public Attributes

- VteTerminal vte_object

  The actual terminal.

5.6.1 Member Data Documentation

5.6.1.1 vte_object

```
VteTerminal _GtkTermTerminal::vte_object
```

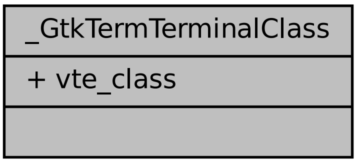
The actual terminal.

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

- terminal.c

5.7 _GtkTermTerminalClass Struct Reference

Collaboration diagram for _GtkTermTerminalClass:



Public Attributes

- VteTerminalClass vte_class

5.7.1 Member Data Documentation

5.7.1.1 vte_class

```
VteTerminalClass _GtkTermTerminalClass::vte_class
```

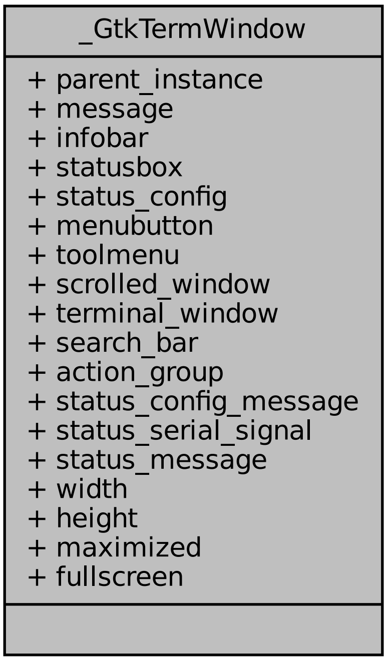
The documentation for this struct was generated from the following file:

- terminal.c

5.8 _GtkTermWindow Struct Reference

The main GtkTermWindow class.

Collaboration diagram for _GtkTermWindow:



Public Attributes

- GtkApplicationWindow parent_instance
- GtkWidget ∗ message

  Message for the infobar.
- GtkWidget ∗ infobar

  Infobar.
- GtkBox ∗ statusbox

  Box for statusbar messages.
- GtkBox ∗ status_config

  Displays the actual used configuration.
- GtkWidget ∗ menubutton

  Toolbar.
- GMenuModel ∗ toolmenu

  Menu.
- GtkScrolledWindow ∗ scrolled_window

  Make the terminal window scrolled.
- GtkTermTerminal ∗ terminal_window

  The terminal window.

- GtkWidget ∗ search_bar

  Searchbar.
- GActionGroup ∗ action_group

  Window action group.
- GtkWidget ∗ status_config_message [3]
- GtkWidget ∗ status_serial_signal [6]
- GtkWidget ∗ status_message
- int width
- int height
- bool maximized
- bool fullscreen

5.8.1 Detailed Description

The main GtkTermWindow class.

MainWindow specific variables here.

5.8.2 Member Data Documentation

5.8.2.1 action_group

```
GActionGroup* _GtkTermWindow::action_group
```

Window action group.

5.8.2.2 fullscreen

```
bool _GtkTermWindow::fullscreen
```

5.8.2.3 height

```
int _GtkTermWindow::height
```

5.8.2.4 infobar

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

Infobar.

5.8.2.5 maximized

```
bool _GtkTermWindow::maximized
```

5.8.2.6 menubutton

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

Toolbar.

5.8.2.7 message

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

Message for the infobar.

5.8.2.8 parent_instance

```
GtkApplicationWindow _GtkTermWindow::parent_instance
```

5.8.2.9 scrolled_window

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

Make the terminal window scrolled.

Referenced by create_window().

5.8.2.10 search_bar

```
GtkWidget* _GtkTermWindow::search_bar
```

Searchbar.

5.8.2.11 status_config

```
GtkBox* _GtkTermWindow::status_config
```

Displays the actual used configuration.

5.8.2.12 status_config_message

```
GtkWidget* _GtkTermWindow::status_config_message[3]
```

5.8.2.13 status_message

```
GtkWidget* _GtkTermWindow::status_message
```

5.8.2.14 status_serial_signal

```
GtkWidget* _GtkTermWindow::status_serial_signal[6]
```

5.8.2.15 statusbox

```
GtkBox* _GtkTermWindow::statusbox
```

Box for statusbar messages.

5.8.2.16 terminal_window

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

The terminal window.

Referenced by create_window().

5.8.2.17 toolmenu

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

Menu.

5.8.2.18 width

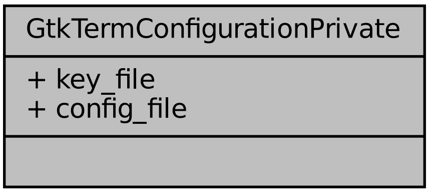
```
int _GtkTermWindow::width
```

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

- gtkterm_window.c

5.9 GtkTermConfigurationPrivate Struct Reference

Collaboration diagram for GtkTermConfigurationPrivate:



Public Attributes

- GKeyFile ∗ key_file

  The memory loaded keyfile.
- GFile ∗ config_file

  The config file.

5.9.1 Member Data Documentation

5.9.1.1 config_file

```
GFile* GtkTermConfigurationPrivate::config_file
```

The config file.

5.9.1.2 key_file

```
GKeyFile* GtkTermConfigurationPrivate::key_file
```

The memory loaded keyfile.

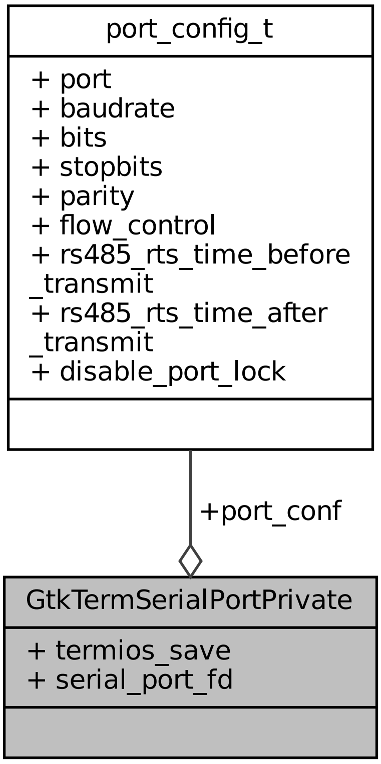
Referenced by check_keyfile(), gtkterm_configuration_default_configuration(), gtkterm_configuration_validate(), and on_set_config_options().

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

- resource_file.c

5.10 GtkTermSerialPortPrivate Struct Reference

Collaboration diagram for GtkTermSerialPortPrivate:



Public Attributes

- port_config_t ∗ port_conf
- struct termios termios_save
- int serial_port_fd

5.10.1 Member Data Documentation

5.10.1.1 port_conf

```
port_config_t* GtkTermSerialPortPrivate::port_conf
```

Referenced by gtkterm_serial_port_get_string().

5.10.1.2 serial_port_fd

```
int GtkTermSerialPortPrivate::serial_port_fd
```

Referenced by gtkterm_serial_port_get_string(), and gtkterm_serial_port_status().

5.10.1.3 termios_save

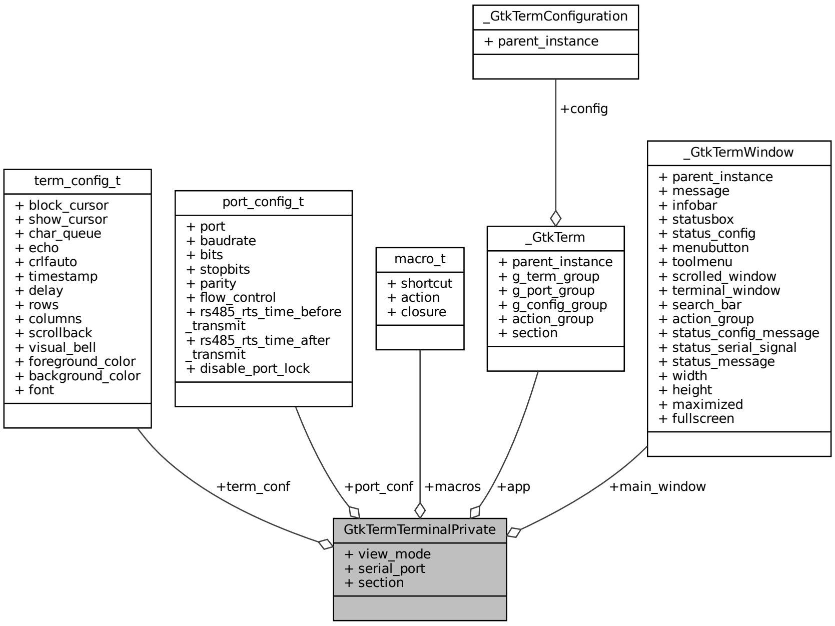
```
struct termios GtkTermSerialPortPrivate::termios_save
```

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

- serial.c

5.11 GtkTermTerminalPrivate Struct Reference

Collaboration diagram for GtkTermTerminalPrivate:



Public Attributes

- uint8_t view_mode

  ASCII or HEX view mode.
- GtkTermSerialPort ∗ serial_port

  The active serial port for this terminal.
- term_config_t ∗ term_conf

  The configuration loaded from the keyfile.
- port_config_t ∗ port_conf

  Port configuration used in this terminal.
- macro_t ∗ macros
- char ∗ section

  Section used in this terminal for configuration from config file.
- GtkTerm ∗ app

  Pointer to the app for getting [section] and keyfile.
- GtkTermWindow ∗ main_window

  Pointer to the main window for updating the statusbar on changes.

5.11.1 Member Data Documentation

5.11.1.1 app

`GtkTerm* GtkTermTerminalPrivate::app`

Pointer to the app for getting [section] and keyfile.

5.11.1.2 macros

`macro_t* GtkTermTerminalPrivate::macros`

Todo convert macros -> object

5.11.1.3 main_window

`GtkTermWindow* GtkTermTerminalPrivate::main_window`

Pointer to the main window for updating the statusbar on changes.

5.11.1.4 port_conf

`port_config_t* GtkTermTerminalPrivate::port_conf`

Port configuration used in this terminal.

5.11.1.5 section

`char* GtkTermTerminalPrivate::section`

Section used in this terminal for configuration from config file.

5.11.1.6 serial_port

`GtkTermSerialPort* GtkTermTerminalPrivate::serial_port`

The active serial port for this terminal.

5.11.1.7 term_conf

```
term_config_t* GtkTermTerminalPrivate::term_conf
```

The configuration loaded from the keyfile.

5.11.1.8 view_mode

```
uint8_t GtkTermTerminalPrivate::view_mode
```

ASCII or HEX view mode.

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

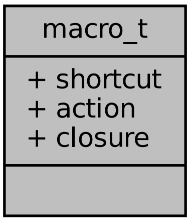
- terminal.c

5.12 macro_t Struct Reference

todo: Migrate to GObject

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

Collaboration diagram for macro_t:



Public Attributes

- char ∗ shortcut

  Shortcut of the macro.
- char ∗ action

  Command to perform.
- GClosure ∗ closure

5.12.1 Detailed Description

todo: Migrate to GObject

Define macro structure type

5.12.2 Member Data Documentation

5.12.2.1 action

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

Command to perform.

Referenced by convert_macros_to_string(), and convert_string_to_macros().

5.12.2.2 closure

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

5.12.2.3 shortcut

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

Shortcut of the macro.

Referenced by convert_macros_to_string(), and convert_string_to_macros().

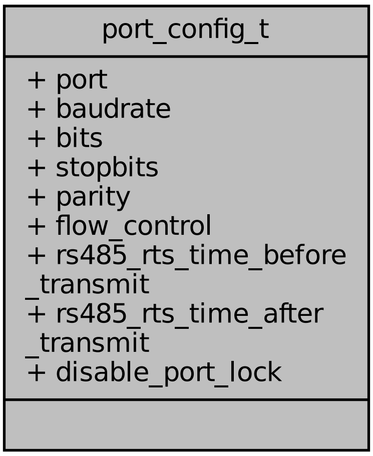
The documentation for this struct was generated from the following file:

- macros.h

5.13 port_config_t Struct Reference

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

Collaboration diagram for port_config_t:



Public Attributes

- char ∗ port
- long int baudrate
- int bits
- int stopbits
- int parity
- int flow_control
- int rs485_rts_time_before_transmit
- int rs485_rts_time_after_transmit
- bool disable_port_lock

5.13.1 Member Data Documentation

5.13.1.1 baudrate

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

Referenced by gtkterm_serial_port_get_string().

5.13.1.2 bits

```
int port_config_t::bits
```

Referenced by gtkterm_serial_port_get_string().

5.13.1.3 disable_port_lock

```
bool port_config_t::disable_port_lock
```

5.13.1.4 flow_control

```
int port_config_t::flow_control
```

5.13.1.5 parity

```
int port_config_t::parity
```

Referenced by gtkterm_serial_port_get_string().

5.13.1.6 port

```
char* port_config_t::port
```

Referenced by gtkterm_serial_port_get_string().

5.13.1.7 rs485_rts_time_after_transmit

```
int port_config_t::rs485_rts_time_after_transmit
```

5.13.1.8 rs485_rts_time_before_transmit

```
int port_config_t::rs485_rts_time_before_transmit
```

5.13.1.9 stopbits

```
int port_config_t::stopbits
```

Referenced by gtkterm_serial_port_get_string().

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

- serial.h

5.14 term_config_t Struct Reference

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

Collaboration diagram for term_config_t:

term_config_t
+ block_cursor
+ show_cursor
+ char_queue
+ echo
+ crlfauto
+ timestamp
+ delay
+ rows
+ columns
+ scrollback
+ visual_bell
+ foreground_color
+ background_color
+ font

Public Attributes

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

5.14.1 Member Data Documentation

5.14.1.1 background_color

```
GdkRGBA term_config_t::background_color
```

5.14.1.2 block_cursor

```
bool term_config_t::block_cursor
```

5.14.1.3 char_queue

```
char term_config_t::char_queue
```

5.14.1.4 columns

```
int term_config_t::columns
```

5.14.1.5 crlfauto

```
bool term_config_t::crlfauto
```

5.14.1.6 delay

```
int term_config_t::delay
```

5.14.1.7 echo

```
bool term_config_t::echo
```

5.14.1.8 font

```
PangoFontDescription* term_config_t::font
```

5.14.1.9 foreground_color

```
GdkRGBA term_config_t::foreground_color
```

5.14.1.10 rows

```
int term_config_t::rows
```

5.14.1.11 scrollback

```
int term_config_t::scrollback
```

5.14.1.12 show_cursor

```
bool term_config_t::show_cursor
```

5.14.1.13 timestamp

```
bool term_config_t::timestamp
```

5.14.1.14 visual_bell

```
bool term_config_t::visual_bell
```

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

- terminal.h

Chapter 6

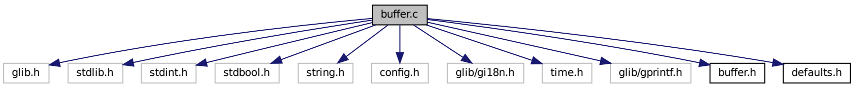
File Documentation

6.1 README_source.md File Reference

6.2 buffer.c File Reference

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

Include dependency graph for buffer.c:



Macros

- #define MAX_SECTION_LENGTH 32
- #define TIMESTAMP_SIZE 50

Functions

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

Variables

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

6.2.1 Macro Definition Documentation

6.2.1.1 MAX_SECTION_LENGTH

```
#define MAX_SECTION_LENGTH 32
```

6.2.1.2 TIMESTAMP_SIZE

```
#define TIMESTAMP_SIZE 50
```

6.2.2 Function Documentation

6.2.2.1 clear_buffer()

```
void clear_buffer (
             void  )
```

References clear_func.

6.2.2.2 create_buffer()

```
void create_buffer (
             void  )
```

6.2.2.3 delete_buffer()

```
void delete_buffer (
             void  )
```

6.2.2.4 insert_timestamp()

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

6.2.2.5 put_chars()

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

References RECEIVE_BUFFER, timestamp_on, and TIMESTAMP_SIZE.

6.2.2.6 set_clear_func()

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

References clear_func.

6.2.2.7 set_display_func()

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

References write_func.

6.2.2.8 unset_clear_func()

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

References clear_func.

6.2.2.9 unset_display_func()

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

References write_func.

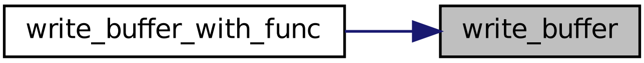
6.2.2.10 write_buffer()

```
void write_buffer (
             void )
```

References overlapped, and write_func.

Referenced by write_buffer_with_func().

Here is the caller graph for this function:

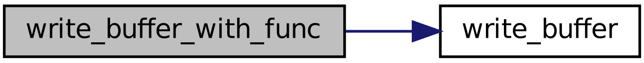


6.2.2.11 write_buffer_with_func()

```
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:



6.2.3 Variable Documentation

6.2.3.1 clear_func

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

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

6.2.3.2 overlapped

```
char overlapped
```

Referenced by write_buffer().

6.2.3.3 timestamp_on

```
bool timestamp_on  [extern]
```

Referenced by put_chars().

6.2.3.4 virt_col_pos

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

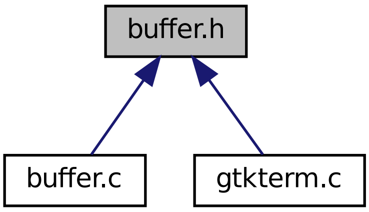
6.2.3.5 write_func

```
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().

6.3 buffer.h File Reference

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



Macros

- #define BUFFER_SIZE (128 ∗ 1024)

Functions

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

6.3.1 Macro Definition Documentation

6.3.1.1 BUFFER_SIZE

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

6.3.2 Function Documentation

6.3.2.1 clear_buffer()

```
void clear_buffer (
             void  )
```

References clear_func.

6.3.2.2 create_buffer()

```
void create_buffer (
             void  )
```

6.3.2.3 delete_buffer()

```
void delete_buffer (
             void  )
```

6.3.2.4 put_chars()

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

References RECEIVE_BUFFER, timestamp_on, and TIMESTAMP_SIZE.

6.3.2.5 set_clear_func()

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

References clear_func.

6.3.2.6 set_display_func()

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

6.3.2.7 unset_clear_func()

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

References clear_func.

6.3.2.8 unset_display_func()

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

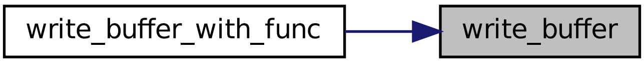
6.3.2.9 write_buffer()

```
void write_buffer (
             void  )
```

References overlapped, and write_func.

Referenced by write_buffer_with_func().

Here is the caller graph for this function:



6.3.2.10 write_buffer_with_func()

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

6.4 buffer.h

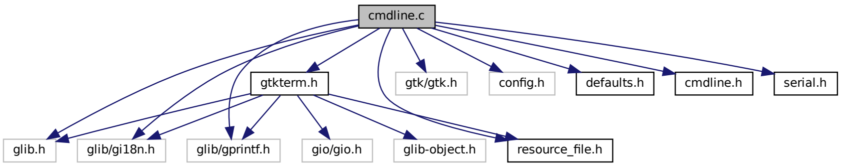
Go to the documentation of this file.

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

#ifndef BUFFER_H_
#define BUFFER_H_

#define BUFFER_SIZE (128 * 1024)

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

#endif
```

6.5 cmdline.c File Reference

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

Include dependency graph for cmdline.c:



Functions

- void gtkterm_add_cmdline_options (GtkTerm *app)

6.5.1 Function Documentation

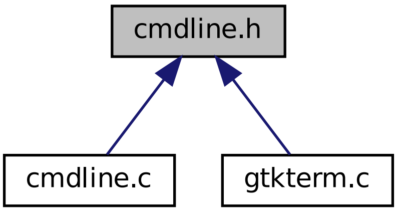
6.5.1.1 gtkterm_add_cmdline_options()

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

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

6.6 cmdline.h File Reference

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



Functions

- void gtkterm_add_cmdline_options (GtkTerm ∗app)

Variables

- GOptionGroup ∗ g_term_group

6.6.1 Function Documentation

6.6.1.1 gtkterm_add_cmdline_options()

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

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

6.6.2 Variable Documentation

6.6.2.1 g_term_group

```
GOptionGroup* g_term_group  [extern]
```

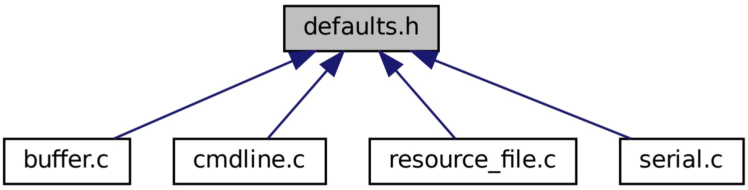
6.7 cmdline.h

Go to the documentation of this file.

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

#ifndef CMDLINE_H
#define CMDLINE_H

void gtkterm_add_cmdline_options (GtkTerm *app);

#endif // CMDLINE_H
```

6.8 defaults.h File Reference

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



Macros

- #define DEFAULT_FONT "Monospace 12"

  Default for VTE-terminal.
- #define DEFAULT_SCROLLBACK 10000
- #define DEFAULT_DELAY 0
- #define DEFAULT_CHAR -1
- #define DEFAULT_DELAY_RS485 30
- #define DEFAULT_ECHO "false"
- #define DEFAULT_VISUAL_BELL "false"
- #define DEFAULT_PORT "/dev/ttyS0"

  Default for serial ports.
- #define DEFAULT_BAUDRATE 115200
- #define DEFAULT_PARITY "none"
- #define DEFAULT_BITS 8
- #define DEFAULT_STOPBITS 1
- #define DEFAULT_FLOW "none"
- #define RECEIVE_BUFFER 8192
- #define TRANSMIT_BUFFER 4096
- #define LINE_FEED 0x0A
- #define POLL_DELAY 100

  in ms (for control signals)
- #define BUFFER_LENGTH 256

  Generic defaults.
- #define MAX_SECTION_LENGTH 32

6.8.1 Macro Definition Documentation

6.8.1.1 BUFFER_LENGTH

```
#define BUFFER_LENGTH 256
```

Generic defaults.

6.8.1.2 DEFAULT_BAUDRATE

```
#define DEFAULT_BAUDRATE 115200
```

6.8.1.3 DEFAULT_BITS

```
#define DEFAULT_BITS 8
```

6.8.1.4 DEFAULT_CHAR

```
#define DEFAULT_CHAR -1
```

6.8.1.5 DEFAULT_DELAY

```
#define DEFAULT_DELAY 0
```

6.8.1.6 DEFAULT_DELAY_RS485

```
#define DEFAULT_DELAY_RS485 30
```

6.8.1.7 DEFAULT_ECHO

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

6.8.1.8 DEFAULT_FLOW

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

6.8.1.9 DEFAULT_FONT

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

Default for VTE-terminal.

6.8.1.10 DEFAULT_PARITY

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

6.8.1.11 DEFAULT_PORT

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

Default for serial ports.

6.8.1.12 DEFAULT_SCROLLBACK

```
#define DEFAULT_SCROLLBACK 10000
```

6.8.1.13 DEFAULT_STOPBITS

```
#define DEFAULT_STOPBITS 1
```

6.8.1.14 DEFAULT_VISUAL_BELL

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

6.8.1.15 LINE_FEED

```
#define LINE_FEED 0x0A
```

6.8.1.16 MAX_SECTION_LENGTH

```
#define MAX_SECTION_LENGTH 32
```

6.8.1.17 POLL_DELAY

```
#define POLL_DELAY 100
```

in ms (for control signals)

6.8.1.18 RECEIVE_BUFFER

```
#define RECEIVE_BUFFER 8192
```

6.8.1.19 TRANSMIT_BUFFER

```
#define TRANSMIT_BUFFER 4096
```

6.9 defaults.h

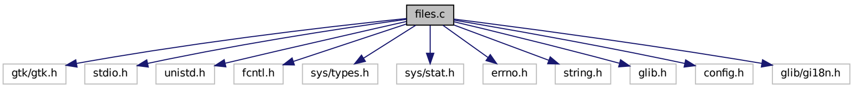
Go to the documentation of this file.

```
//!  Default for VTE-terminal
#define DEFAULT_FONT            "Monospace 12"
#define DEFAULT_SCROLLBACK      10000
#define DEFAULT_DELAY           0
#define DEFAULT_CHAR            -1
#define DEFAULT_DELAY_RS485     30
#define DEFAULT_ECHO            "false"
#define DEFAULT_VISUAL_BELL     "false"

//!  Default for serial ports
#define DEFAULT_PORT            "/dev/ttyS0"
#define DEFAULT_BAUDRATE        115200
#define DEFAULT_PARITY          "none"
#define DEFAULT_BITS            8
#define DEFAULT_STOPBITS        1
#define DEFAULT_FLOW            "none"

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

//!  Generic defaults
#define BUFFER_LENGTH           256
#define MAX_SECTION_LENGTH      32
```

6.10 files.c File Reference

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

Include dependency graph for files.c:



Variables

- char ∗ default_filename = NULL

6.10.1 Variable Documentation

6.10.1.1 default_filename

```
char* default_filename = NULL
```

6.11 files.h File Reference

Functions

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

Variables

- gboolean waiting_for_char
- char ∗ default_filename

6.11.1 Function Documentation

6.11.1.1 add_input()

```
void add_input (
             void  )
```

6.11.1.2 save_raw_file()

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

6.11.1.3 send_raw_file()

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

6.11.2 Variable Documentation

6.11.2.1 default_filename

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

6.11.2.2 waiting_for_char

```
gboolean waiting_for_char  [extern]
```

6.12 files.h

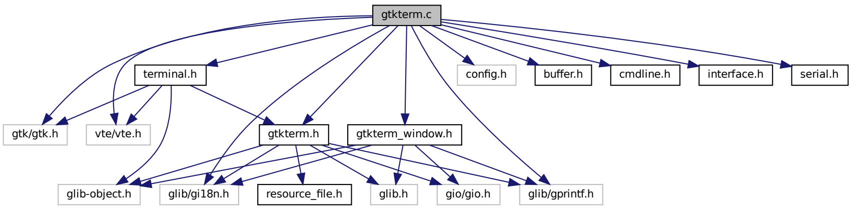
Go to the documentation of this file.

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

#ifndef FILES_H_
#define FILES_H_

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

extern gboolean waiting_for_char;
extern char *default_filename;

#endif
```

6.13 gtkterm.c File Reference

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

Include dependency graph for gtkterm.c:



Functions

- int main (int argc, char ∗argv[ ])

Variables

- unsigned int gtkterm_signals [LAST_GTKTERM_SIGNAL]

6.13.1 Function Documentation

6.13.1.1 main()

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

References GTKTERM_TYPE_APP.

6.13.2 Variable Documentation

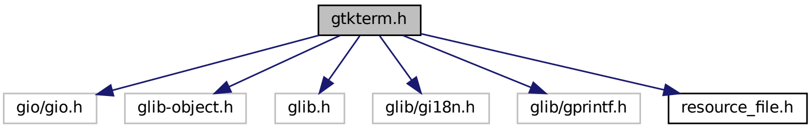
6.13.2.1 gtkterm_signals

```
unsigned int gtkterm_signals[LAST_GTKTERM_SIGNAL]
```

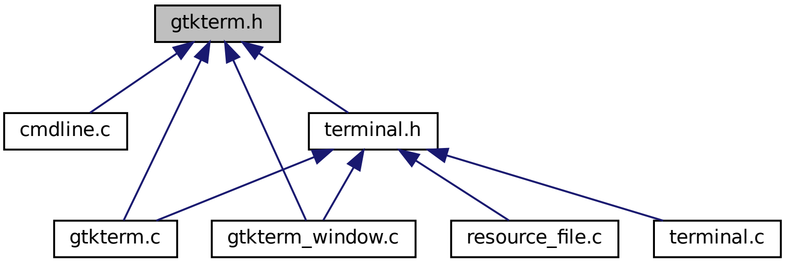
6.14 gtkterm.h File Reference

```
#include <gio/gio.h>
#include <glib-object.h>
#include <glib.h>
#include <glib/gi18n.h>
#include <glib/gprintf.h>
#include "resource_file.h"
```

Include dependency graph for gtkterm.h:



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



Classes

- struct _GtkTerm

  The main GtkTerm application class.

Macros

- #define GTKTERM_TYPE_APP gtkterm_get_type()

Typedefs

- typedef struct _GtkTerm GtkTerm

Enumerations

- enum {
SIGNAL_GTKTERM_LOAD_CONFIG , SIGNAL_GTKTERM_SAVE_CONFIG , SIGNAL_GTKTERM_REMOVE_SECTION , SIGNAL_GTKTERM_PRINT_SECTION ,
SIGNAL_GTKTERM_COPY_SECTION , SIGNAL_GTKTERM_CONFIG_TERMINAL , SIGNAL_GTKTERM_CONFIG_SERIAL , SIGNAL_GTKTERM_TERMINAL_CHANGED ,
LAST_GTKTERM_SIGNAL }

Variables

- unsigned int gtkterm_signals [ ]

6.14.1 Macro Definition Documentation

6.14.1.1 GTKTERM_TYPE_APP

```
#define GTKTERM_TYPE_APP gtkterm_get_type()
```

6.14.2 Typedef Documentation

6.14.2.1 GtkTerm

```
typedef struct _GtkTerm GtkTerm
```

6.14.3 Enumeration Type Documentation

6.14.3.1 anonymous enum

```
anonymous enum
```

Enumerator

SIGNAL_GTKTERM_LOAD_CONFIG	
SIGNAL_GTKTERM_SAVE_CONFIG	
SIGNAL_GTKTERM_REMOVE_SECTION	
SIGNAL_GTKTERM_PRINT_SECTION	
SIGNAL_GTKTERM_COPY_SECTION	
SIGNAL_GTKTERM_CONFIG_TERMINAL	
SIGNAL_GTKTERM_CONFIG_SERIAL	
SIGNAL_GTKTERM_TERMINAL_CHANGED	
LAST_GTKTERM_SIGNAL	

6.14.4 Variable Documentation

6.14.4.1 gtkterm_signals

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

6.15 gtkterm.h

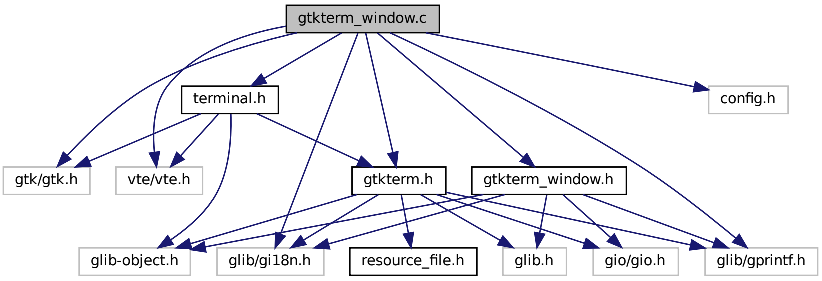
Go to the documentation of this file.

```

#ifndef GTKTERM_H
#define GTKTERM_H

#include <gio/gio.h>
#include <glib-object.h>
#include <glib.h>
#include <glib/gi18n.h>
#include <glib/gprintf.h>

#include "resource_file.h"

enum {
    SIGNAL_GTKTERM_LOAD_CONFIG,
    SIGNAL_GTKTERM_SAVE_CONFIG,
    SIGNAL_GTKTERM_REMOVE_SECTION,
    SIGNAL_GTKTERM_PRINT_SECTION,
    SIGNAL_GTKTERM_COPY_SECTION,
    SIGNAL_GTKTERM_CONFIG_TERMINAL,
    SIGNAL_GTKTERM_CONFIG_SERIAL,
    SIGNAL_GTKTERM_TERMINAL_CHANGED,
    LAST_GTKTERM_SIGNAL
};

extern unsigned int gtkterm_signals[];

G_BEGIN_DECLS

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

  GtkApplication parent_instance;

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

  GActionGroup *action_group;           //!< App action group

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

#define GTKTERM_TYPE_APP gtkterm_get_type()
typedef struct _GtkTerm GtkTerm;
G_DECLARE_FINAL_TYPE (GtkTerm, gtkterm, GTKTERM, APP, GtkApplication)


#endif // GTKTERM_H
```

6.16 gtkterm_window.c File Reference

```
#include <gtk/gtk.h>
#include <vte/vte.h>
#include <glib/gi18n.h>
#include <glib/gprintf.h>
#include "config.h"
#include "gtkterm.h"
#include "gtkterm_window.h"
#include "terminal.h"
```

Include dependency graph for gtkterm_window.c:



Classes

- struct _GtkTermWindow

  The main GtkTermWindow class.

Functions

- void set_window_title (GtkTermWindow *, gpointer)
- void create_window (GApplication *app)

6.16.1 Function Documentation

6.16.1.1 create_window()

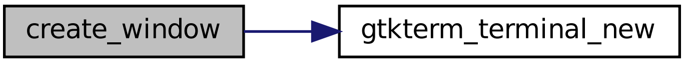
```
void create_window (
            GApplication * app )
```

Create a new terminal window and send section and keyfile as parameter GTKTERM_TERMINAL then can load the right section.

Make the VTE window scrollable

References gtkterm_terminal_new(), _GtkTermWindow::scrolled_window, and _GtkTermWindow::terminal_window.

Here is the call graph for this function:



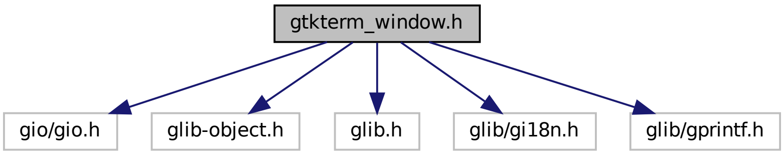
6.16.1.2 set_window_title()

```
void set_window_title (
            GtkTermWindow * window,
            gpointer serial_config_string )
```

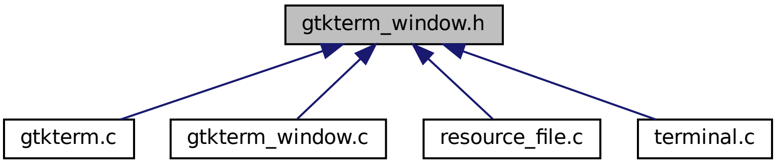
6.17 gtkterm_window.h File Reference

```
#include <gio/gio.h>
#include <glib-object.h>
#include <glib.h>
#include <glib/gi18n.h>
#include <glib/gprintf.h>
```

Include dependency graph for gtkterm_window.h:



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



Macros

- #define GTKTERM_TYPE_GTKTERM_WINDOW gtkterm_window_get_type()

Typedefs

- typedef struct _GtkTermWindow GtkTermWindow

Functions

- G_END_DECLS void create_window (GApplication ∗)

6.17.1 Macro Definition Documentation

6.17.1.1 GTKTERM_TYPE_GTKTERM_WINDOW

```
#define GTKTERM_TYPE_GTKTERM_WINDOW gtkterm_window_get_type()
```

6.17.2 Typedef Documentation

6.17.2.1 GtkTermWindow

```
typedef struct _GtkTermWindow GtkTermWindow
```

6.17.3 Function Documentation

6.17.3.1 create_window()

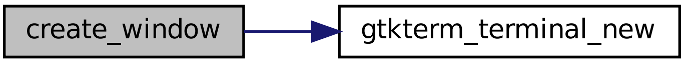
```
G_END_DECLS void create_window (
             GApplication * app )
```

Create a new terminal window and send section and keyfile as parameter GTKTERM_TERMINAL then can load the right section.

Make the VTE window scrollable

References gtkterm_terminal_new(), _GtkTermWindow::scrolled_window, and _GtkTermWindow::terminal_window.

Here is the call graph for this function:



6.18 gtkterm_window.h

Go to the documentation of this file.

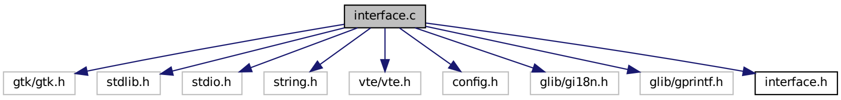
```
#include <gio/gio.h>
#include <glib-object.h>
#include <glib.h>
#include <glib/gi18n.h>
#include <glib/gprintf.h>

#ifndef GTKTERM_WINDOW_H
#define GTKTERM_WINDOW_H

G_BEGIN_DECLS

#define GTKTERM_TYPE_GTKTERM_WINDOW gtkterm_window_get_type()
typedef struct _GtkTermWindow GtkTermWindow;
G_DECLARE_FINAL_TYPE (GtkTermWindow, gtkterm_window, GTKTERM, WINDOW, GtkApplicationWindow)

G_END_DECLS

void create_window (GApplication *);

#endif // GTKTERM_WINDOW_H
```

6.19 interface.c File Reference

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

`#include "interface.h"`

Include dependency graph for interface.c:



Functions

- void show_message (char ∗message, int type_msg)

Variables

- bool timestamp_on = 0
- int virt_col_pos = 0

6.19.1 Function Documentation

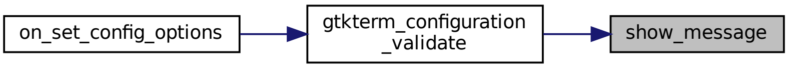
6.19.1.1 show_message()

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

References MSG_ERR.

Referenced by gtkterm_configuration_validate().

Here is the caller graph for this function:



6.19.2 Variable Documentation

6.19.2.1 timestamp_on

```
bool timestamp_on = 0
```

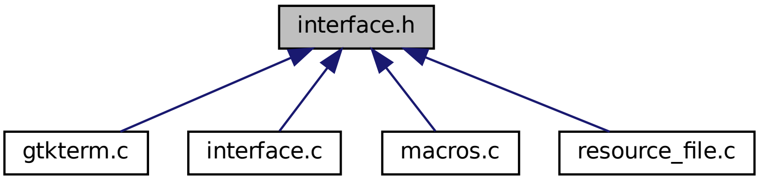
Referenced by put_chars().

6.19.2.2 virt_col_pos

```
int virt_col_pos = 0
```

6.20 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

6.20.1 Macro Definition Documentation

6.20.1.1 ASCII_VIEW

```
#define ASCII_VIEW 0
```

6.20.1.2 HEXADECIMAL_VIEW

```
#define HEXADECIMAL_VIEW 1
```

6.20.1.3 MSG_ERR

```
#define MSG_ERR 1
```

6.20.1.4 MSG_WRN

```
#define MSG_WRN 0
```

6.20.2 Function Documentation

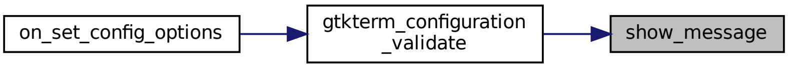
6.20.2.1 show_message()

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

References MSG_ERR.

Referenced by gtkterm_configuration_validate().

Here is the caller graph for this function:



6.20.3 Variable Documentation

6.20.3.1 display

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

6.20.3.2 Text

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

6.21 interface.h

Go to the documentation of this file.

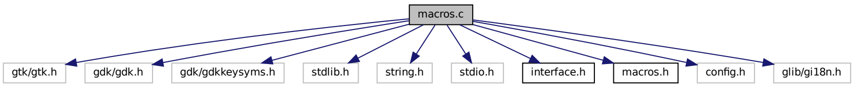
```
/***********************************************************************/
/* interface.h                                                         */
/* ---------                                                           */
/*           GTKTerm Software                                          */
/*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

#ifndef WIDGETS_H_
#define WIDGETS_H_

#define MSG_WRN          0
#define MSG_ERR          1

#define ASCII_VIEW       0
#define HEXADECIMAL_VIEW 1

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

void show_message(char *, int);

#endif
```

6.22 macros.c File Reference

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

`#include <glib/gi18n.h>`

Include dependency graph for macros.c:



Enumerations

- enum { COLUMN_SHORTCUT , COLUMN_ACTION , NUM_COLUMNS }

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)

  Remove shortcuts from accel_group and free memory.

Variables

- macro_t ∗ macros = NULL
- int nr_of_macros = 0

6.22.1 Enumeration Type Documentation

6.22.1.1 anonymous enum

`anonymous enum`

Todo : Migrate to GObject

Enumerator

COLUMN_SHORTCUT	
COLUMN_ACTION	
NUM_COLUMNS	

6.22.2 Function Documentation

6.22.2.1 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.

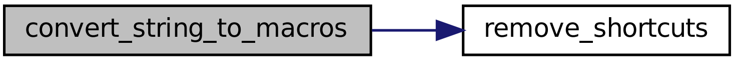
6.22.2.2 convert_string_to_macros()

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

Convert the array of strings to macros.

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

Here is the call graph for this function:



6.22.2.3 get_shortcuts()

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

References macros.

6.22.2.4 macro_count()

```
int macro_count ( )
```

References nr_of_macros.

6.22.2.5 remove_shortcuts()

```
void remove_shortcuts (
             void  )
```

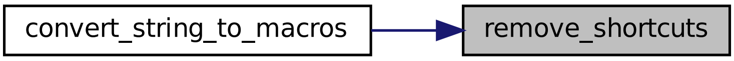
Remove shortcuts from accel_group and free memory.

Clean up all macros

References macros.

Referenced by convert_string_to_macros().

Here is the caller graph for this function:



6.22.3 Variable Documentation

6.22.3.1 macros

```
macro_t* macros = NULL
```

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

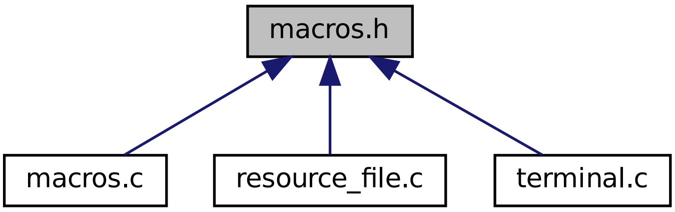
6.22.3.2 nr_of_macros

```
int nr_of_macros = 0
```

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

6.23 macros.h File Reference

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



Classes

- struct macro_t

    todo: Migrate to GObject

Functions

- void remove_shortcuts (void)

    Remove shortcuts from accel_group and free memory.
- void add_shortcuts (void)
- 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

6.23.1 Function Documentation

6.23.1.1 add_shortcuts()

```
void add_shortcuts (
            void   )
```

6.23.1.2 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.

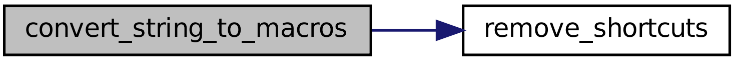
6.23.1.3 convert_string_to_macros()

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

Convert the array of strings to macros.

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

Here is the call graph for this function:



6.23.1.4 get_shortcuts()

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

6.23.1.5 macro_count()

```
int macro_count ( )
```

References nr_of_macros.

6.23.1.6 remove_shortcuts()

```
void remove_shortcuts (
             void  )
```

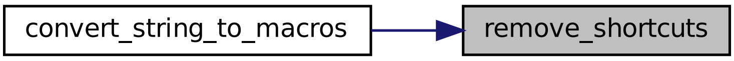
Remove shortcuts from accel_group and free memory.

Clean up all macros

References macros.

Referenced by convert_string_to_macros().

Here is the caller graph for this function:



6.23.2 Variable Documentation

6.23.2.1 macros

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

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

6.24 macros.h

Go to the documentation of this file.

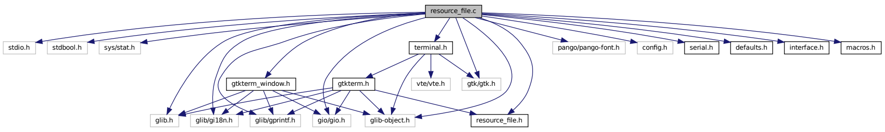
```
/***********************************************************************
* macros.h
* --------
*           GTKTerm Software
*                      (c) Julien Schmitt
*
* -------------------------------------------------------------------
*
* \brief Purpose
*        Functions for the management of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

#ifndef MACROS_H_
#define MACROS_H_

//!  todo:  Migrate to GObject

//!  Define macro structure type
typedef struct
```

```
{
    char *shortcut;     //!< Shortcut of the macro
    char *action;     //!< Command to perform
    GClosure *closure;   //!<
}
macro_t;

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

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

int macro_count ();

extern macro_t *macros;

#endif
```

6.25 resource_file.c File Reference

```
#include <stdio.h>
#include <stdbool.h>
#include <sys/stat.h>
#include <glib.h>
#include <glib/gi18n.h>
#include <glib/gprintf.h>
#include <glib-object.h>
#include <gtk/gtk.h>
#include <gio/gio.h>
#include <pango/pango-font.h>
#include "config.h"
#include "gtkterm_window.h"
#include "serial.h"
#include "terminal.h"
#include "defaults.h"
#include "resource_file.h"
#include "interface.h"
#include "macros.h"
```

Include dependency graph for resource_file.c:



Classes

- struct GtkTermConfigurationPrivate
- struct _GtkTermConfiguration
- struct _GtkTermConfigurationClass

Macros

- #define CONFIGURATION_FILENAME ".gtktermrc"

  Default configuration filename.
- #define BUFFER_LENGTH 256

  Bufferlength for strings.

Functions

- void gtkterm_configuration_default_configuration (GtkTermConfiguration ∗self, char ∗section)

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

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

  Set the config option in the keyfile.

Variables

- const char GtkTermConfigurationItems [ ][CONF_ITEM_LENGTH]

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

6.25.1 Macro Definition Documentation

6.25.1.1 BUFFER_LENGTH

```
#define BUFFER_LENGTH 256
```

Bufferlength for strings.

6.25.1.2 CONFIGURATION_FILENAME

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

Default configuration filename.

Name of the resource file

6.25.2 Function Documentation

6.25.2.1 check_keyfile()

```
int check_keyfile (
            GtkTermConfiguration ∗ self,
            char ∗ section )
```

Load keyfile if it is nog loaded yet

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

References GtkTermConfigurationPrivate::key_file.

6.25.2.2 gtkterm_configuration_default_configuration()

```
void gtkterm_configuration_default_configuration (
            GtkTermConfiguration * self,
            char * section )
```

Create a new <default> configuration.

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

6.25.2.3 gtkterm_configuration_validate()

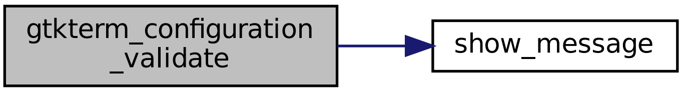
```
void gtkterm_configuration_validate (
            GtkTermConfiguration * self,
            char * section )
```

validate the configuration, given by the section.

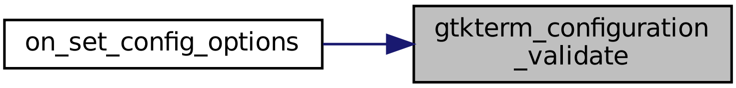
References CONF_ITEM_SERIAL_BAUDRATE, CONF_ITEM_SERIAL_BITS, CONF_ITEM_SERIAL_STOPBITS, CONF_ITEM_TERM_FONT, CONF_ITEM_TERM_WAIT_DELAY, DEFAULT_BITS, DEFAULT_DELAY, DEFAULT_FONT, DEFAULT_STOPBITS, GtkTermConfigurationItems, GtkTermConfigurationPrivate::key_file, MSG_ERR, and show_message().

Referenced by on_set_config_options().

Here is the call graph for this function:



Here is the caller graph for this function:



6.25.2.4 on_set_config_options()

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

Set the config option in the keyfile.

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

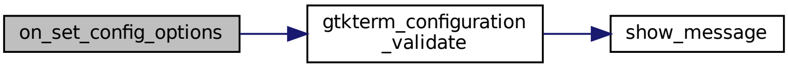
Search index for the option we want to set

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

We should not get here.

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

Here is the call graph for this function:



6.25.3 Variable Documentation

6.25.3.1 GtkTermConfigurationItems

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

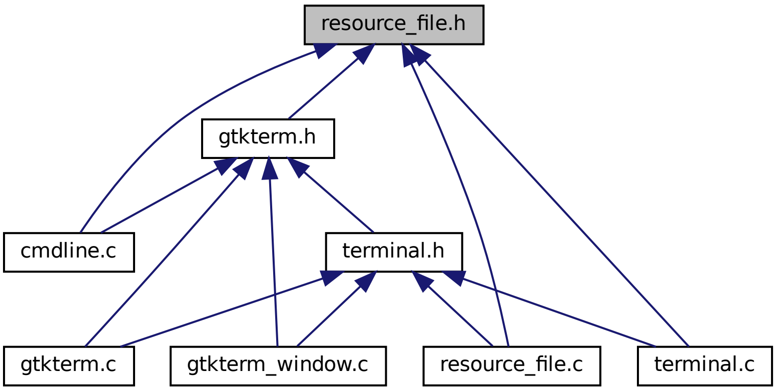
Used configuration options to hold consistency between load/save functions.

Configuration item names.

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

6.26 resource_file.h File Reference

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



Macros

- #define CONF_ITEM_LENGTH 32
- #define DEFAULT_SECTION "default"

  Default section if not specified.
- #define GTKTERM_TYPE_CONFIGURATION gtkterm_configuration_get_type ()

Typedefs

- typedef struct _GtkTermConfiguration GtkTermConfiguration

Enumerations

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

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

Functions

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

  Set the config option in the keyfile.

Variables

- const char GtkTermConfigurationItems [ ][CONF_ITEM_LENGTH]

  Configuration item names.

6.26.1 Macro Definition Documentation

6.26.1.1 CONF_ITEM_LENGTH

```
#define CONF_ITEM_LENGTH 32
```

6.26.1.2 DEFAULT_SECTION

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

Default section if not specified.

6.26.1.3 GTKTERM_TYPE_CONFIGURATION

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

6.26.2 Typedef Documentation

6.26.2.1 GtkTermConfiguration

```
typedef struct _GtkTermConfiguration GtkTermConfiguration
```

6.26.3 Enumeration Type Documentation

6.26.3.1 anonymous enum

```
anonymous enum
```

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

it is an index to ConfigurationItem.

Enumerator

CONF_ITEM_SERIAL_PORT	
CONF_ITEM_SERIAL_BAUDRATE	
CONF_ITEM_SERIAL_BITS	
CONF_ITEM_SERIAL_STOPBITS	
CONF_ITEM_SERIAL_PARITY	
CONF_ITEM_SERIAL_FLOW_CONTROL	
CONF_ITEM_TERM_WAIT_DELAY	
CONF_ITEM_TERM_WAIT_CHAR	
CONF_ITEM_SERIAL_RS485_RTS_TIME_BEFORE_TX	
CONF_ITEM_SERIAL_RS485_RTS_TIME_AFTER_TX	
CONF_ITEM_TERM_MACROS	
CONF_ITEM_TERM_RAW_FILENAME	
CONF_ITEM_TERM_ECHO	
CONF_ITEM_TERM_CRLF_AUTO	
CONF_ITEM_SERIAL_DISABLE_PORT_LOCK	
CONF_ITEM_TERM_FONT	
CONF_ITEM_TERM_TIMESTAMP	
CONF_ITEM_TERM_BLOCK_CURSOR	
CONF_ITEM_TERM_SHOW_CURSOR	
CONF_ITEM_TERM_ROWS	
CONF_ITEM_TERM_COLS	
CONF_ITEM_TERM_SCROLLBACK	

Enumerator

CONF_ITEM_TERM_VISUAL_BELL	
CONF_ITEM_TERM_FOREGROUND_RED	
CONF_ITEM_TERM_FOREGROUND_GREEN	
CONF_ITEM_TERM_FOREGROUND_BLUE	
CONF_ITEM_TERM_FOREGROUND_ALPHA	
CONF_ITEM_TERM_BACKGROUND_RED	
CONF_ITEM_TERM_BACKGROUND_GREEN	
CONF_ITEM_TERM_BACKGROUND_BLUE	
CONF_ITEM_TERM_BACKGROUND_ALPHA	
CONF_ITEM_LAST	Checking as last item in the list.

6.26.4 Function Documentation

6.26.4.1 gtkterm_configuration_new()

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

6.26.4.2 on_set_config_options()

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

Set the config option in the keyfile.

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

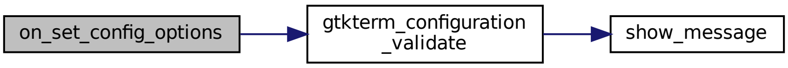
Search index for the option we want to set

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

We should not get here.

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

Here is the call graph for this function:



6.26.5 Variable Documentation

6.26.5.1 GtkTermConfigurationItems

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

Configuration item names.

Configuration item names.

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

6.27 resource_file.h

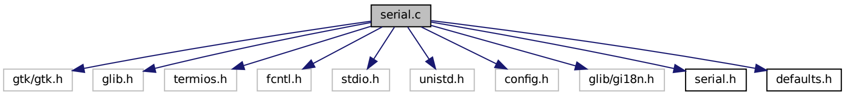
Go to the documentation of this file.

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

#ifndef RESOURCE_FILE_H_
#define RESOURCE_FILE_H_

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

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

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

G_BEGIN_DECLS

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

GtkTermConfiguration *gtkterm_configuration_new (void);

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

G_END_DECLS

#endif
```

6.28 serial.c File Reference

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

Include dependency graph for serial.c:



Classes

- struct GtkTermSerialPortPrivate
- struct _GtkTermSerialPort
- struct _GtkTermSerialPortClass

Enumerations

- enum { PROP_0 , PROP_PORT_CONFIG , N_PROPS }

Functions

- GtkTermSerialPort ∗ gtkterm_serial_port_new (port_config_t ∗port_conf)
- char ∗ gtkterm_serial_port_get_string (GtkTermSerialPort ∗self)
- int gtkterm_serial_port_status (GtkTermSerialPort ∗self)

6.28.1 Enumeration Type Documentation

6.28.1.1 anonymous enum

```
anonymous enum
```

Enumerator

PROP_0	
PROP_PORT_CONFIG	
N_PROPS	

6.28.2 Function Documentation

6.28.2.1 gtkterm_serial_port_get_string()

```
char * gtkterm_serial_port_get_string (
            GtkTermSerialPort * self )
```

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

6.28.2.2 gtkterm_serial_port_new()

```
GtkTermSerialPort * gtkterm_serial_port_new (
            port_config_t * port_conf )
```

References GTKTERM_TYPE_SERIAL_PORT.

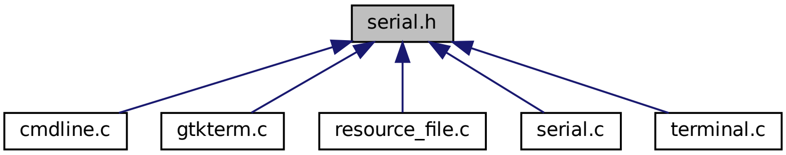
6.28.2.3 gtkterm_serial_port_status()

```
int gtkterm_serial_port_status (
            GtkTermSerialPort * self )
```

References GtkTermSerialPortPrivate::serial_port_fd.

6.29 serial.h File Reference

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



Classes

- struct port_config_t

Macros

- #define GTKTERM_TYPE_SERIAL_PORT gtkterm_serial_port_get_type ()

Typedefs

- typedef typedefG_BEGIN_DECLS struct _GtkTermSerialPort GtkTermSerialPort

Functions

- GtkTermSerialPort ∗ gtkterm_serial_port_new (port_config_t ∗)
- G_END_DECLS char ∗ gtkterm_serial_port_get_string (GtkTermSerialPort ∗)
- int gtkterm_serial_port_status (GtkTermSerialPort ∗)

6.29.1 Macro Definition Documentation

6.29.1.1 GTKTERM_TYPE_SERIAL_PORT

```
#define GTKTERM_TYPE_SERIAL_PORT gtkterm_serial_port_get_type ()
```

6.29.2 Typedef Documentation

6.29.2.1 GtkTermSerialPort

```
typedef typedefG_BEGIN_DECLS struct _GtkTermSerialPort GtkTermSerialPort
```

6.29.3 Function Documentation

6.29.3.1 gtkterm_serial_port_get_string()

```
G_END_DECLS char * gtkterm_serial_port_get_string (
            GtkTermSerialPort * self )
```

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

6.29.3.2 gtkterm_serial_port_new()

```
GtkTermSerialPort * gtkterm_serial_port_new (
            port_config_t * port_conf )
```

References GTKTERM_TYPE_SERIAL_PORT.

6.29.3.3 gtkterm_serial_port_status()

```
int gtkterm_serial_port_status (
            GtkTermSerialPort * self )
```

References GtkTermSerialPortPrivate::serial_port_fd.

6.30 serial.h

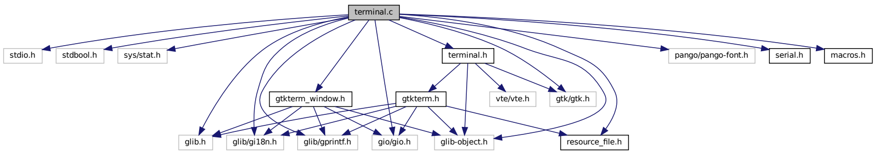
Go to the documentation of this file.

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

#ifndef SERIAL_H_
#define SERIAL_H_

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

} port_config_t;

G_BEGIN_DECLS

typedef struct _GtkTermSerialPort GtkTermSerialPort;

#define GTKTERM_TYPE_SERIAL_PORT gtkterm_serial_port_get_type ()
G_DECLARE_FINAL_TYPE (GtkTermSerialPort, gtkterm_serial_port, GTKTERM, SERIAL_PORT, GObject)

GtkTermSerialPort *gtkterm_serial_port_new (port_config_t *);

G_END_DECLS

char* gtkterm_serial_port_get_string (GtkTermSerialPort *);
int gtkterm_serial_port_status (GtkTermSerialPort *);

#endif
```

6.31 terminal.c File Reference

```c
#include <stdio.h>
#include <stdbool.h>
#include <sys/stat.h>
#include <glib.h>
#include <glib/gi18n.h>
#include <glib/gprintf.h>
#include <glib-object.h>
#include <gtk/gtk.h>
#include <gio/gio.h>
#include <pango/pango-font.h>
#include "gtkterm_window.h"
#include "terminal.h"
#include "serial.h"
#include "macros.h"
#include "resource_file.h"
```

Include dependency graph for terminal.c:



Classes

- struct GtkTermTerminalPrivate
- struct _GtkTermTerminal
- struct _GtkTermTerminalClass

Enumerations

- enum {
  PROP_0 , PROP_SECTION , PROP_GTKTERM_APP , PROP_MAIN_WINDOW ,
  N_PROPS }

Functions

- GtkTermTerminal ∗ gtkterm_terminal_new (char ∗section, GtkTerm ∗gtkterm_app, GtkTermWindow ∗main←
  _window)

6.31.1 Enumeration Type Documentation

6.31.1.1 anonymous enum

```
anonymous enum
```

Enumerator

PROP_0	
PROP_SECTION	
PROP_GTKTERM_APP	
PROP_MAIN_WINDOW	
N_PROPS	

6.31.2 Function Documentation

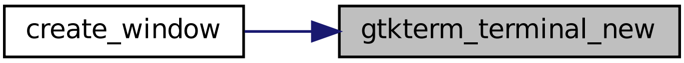
6.31.2.1 gtkterm_terminal_new()

```
GtkTermTerminal * gtkterm_terminal_new (
              char * section,
              GtkTerm * gtkterm_app,
              GtkTermWindow * main_window )
```

References GTKTERM_TYPE_TERMINAL.

Referenced by create_window().

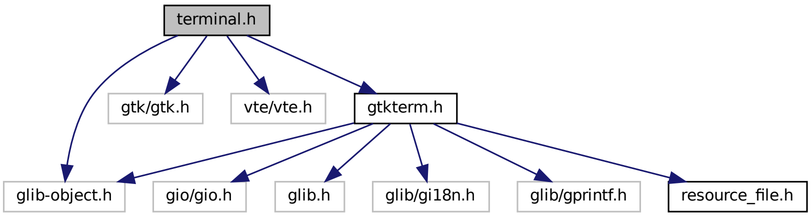
Here is the caller graph for this function:



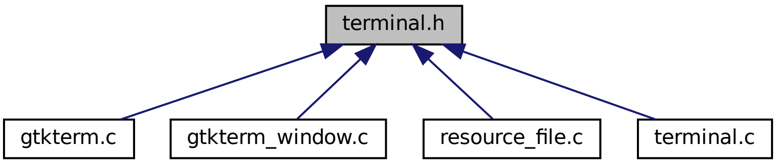
6.32 terminal.h File Reference

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

Include dependency graph for terminal.h:



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



Classes

- struct term_config_t

Macros

- #define GTKTERM_TYPE_TERMINAL gtkterm_terminal_get_type()

Functions

- GtkTermTerminal ∗ gtkterm_terminal_new (char ∗, GtkTerm ∗, GtkTermWindow ∗)

6.32.1 Macro Definition Documentation

6.32.1.1 GTKTERM_TYPE_TERMINAL

```
#define GTKTERM_TYPE_TERMINAL gtkterm_terminal_get_type()
```

6.32.2 Function Documentation

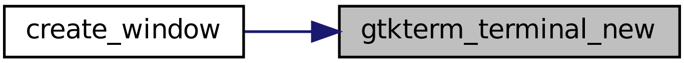
6.32.2.1 gtkterm_terminal_new()

```
GtkTermTerminal * gtkterm_terminal_new (
             char * section,
             GtkTerm * gtkterm_app,
             GtkTermWindow * main_window )
```

References GTKTERM_TYPE_TERMINAL.

Referenced by create_window().

Here is the caller graph for this function:



6.33 terminal.h

Go to the documentation of this file.

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

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

#include "gtkterm.h"

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

} term_config_t;

G_BEGIN_DECLS

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

GtkTermTerminal *gtkterm_terminal_new (char *, GtkTerm *, GtkTermWindow *);

G_END_DECLS

#endif // TERMINAL_H
```

Index

_GtkTerm, 9
    action_group, 10
    config, 10
    g_config_group, 10
    g_port_group, 10
    g_term_group, 11
    parent_instance, 11
    section, 11
_GtkTermConfiguration, 11
    parent_instance, 12
_GtkTermConfigurationClass, 12
    parent_class, 12
_GtkTermSerialPort, 13
    parent_instance, 13
_GtkTermSerialPortClass, 13
    parent_class, 14
_GtkTermTerminal, 14
    vte_object, 14
_GtkTermTerminalClass, 15
    vte_class, 15
_GtkTermWindow, 16
    action_group, 17
    fullscreen, 17
    height, 17
    infobar, 17
    maximized, 17
    menubutton, 18
    message, 18
    parent_instance, 18
    scrolled_window, 18
    search_bar, 18
    status_config, 18
    status_config_message, 19
    status_message, 19
    status_serial_signal, 19
    statusbox, 19
    terminal_window, 19
    toolmenu, 19
    width, 20

action
    macro_t, 26
action_group
    _GtkTerm, 10
    _GtkTermWindow, 17
add_input
    files.h, 48
add_shortcuts
    macros.h, 65
app
    GtkTermTerminalPrivate, 23
ASCII_VIEW
    interface.h, 59

background_color
    term_config_t, 30
baudrate
    port_config_t, 27
bits
    port_config_t, 27
block_cursor
    term_config_t, 30
buffer.c, 33
    clear_buffer, 34
    clear_func, 37
    create_buffer, 34
    delete_buffer, 35
    insert_timestamp, 35
    MAX_SECTION_LENGTH, 34
    overlapped, 37
    put_chars, 35
    set_clear_func, 35
    set_display_func, 35
    timestamp_on, 37
    TIMESTAMP_SIZE, 34
    unset_clear_func, 35
    unset_display_func, 36
    virt_col_pos, 37
    write_buffer, 36
    write_buffer_with_func, 36
    write_func, 37
buffer.h, 38, 41
    BUFFER_SIZE, 38
    clear_buffer, 38
    create_buffer, 39
    delete_buffer, 39
    put_chars, 39
    set_clear_func, 39
    set_display_func, 39
    unset_clear_func, 39
    unset_display_func, 40
    write_buffer, 40
    write_buffer_with_func, 40
BUFFER_LENGTH
    defaults.h, 44
    resource_file.c, 69
BUFFER_SIZE
    buffer.h, 38

char_queue

term_config_t, 30
check_keyfile
resource_file.c, 69
clear_buffer
buffer.c, 34
buffer.h, 38
clear_func
buffer.c, 37
closure
macro_t, 26
cmdline.c, 41
gtkterm_add_cmdline_options, 42
cmdline.h, 42, 43
g_term_group, 43
gtkterm_add_cmdline_options, 42
COLUMN_ACTION
macros.c, 62
COLUMN_SHORTCUT
macros.c, 62
columns
term_config_t, 30
CONF_ITEM_LAST
resource_file.h, 75
CONF_ITEM_LENGTH
resource_file.h, 73
CONF_ITEM_SERIAL_BAUDRATE
resource_file.h, 74
CONF_ITEM_SERIAL_BITS
resource_file.h, 74
CONF_ITEM_SERIAL_DISABLE_PORT_LOCK
resource_file.h, 74
CONF_ITEM_SERIAL_FLOW_CONTROL
resource_file.h, 74
CONF_ITEM_SERIAL_PARITY
resource_file.h, 74
CONF_ITEM_SERIAL_PORT
resource_file.h, 74
CONF_ITEM_SERIAL_RS485_RTS_TIME_AFTER_TX
resource_file.h, 74
CONF_ITEM_SERIAL_RS485_RTS_TIME_BEFORE_TX
resource_file.h, 74
CONF_ITEM_SERIAL_STOPBITS
resource_file.h, 74
CONF_ITEM_TERM_BACKGROUND_ALPHA
resource_file.h, 75
CONF_ITEM_TERM_BACKGROUND_BLUE
resource_file.h, 75
CONF_ITEM_TERM_BACKGROUND_GREEN
resource_file.h, 75
CONF_ITEM_TERM_BACKGROUND_RED
resource_file.h, 75
CONF_ITEM_TERM_BLOCK_CURSOR
resource_file.h, 74
CONF_ITEM_TERM_COLS
resource_file.h, 74
CONF_ITEM_TERM_CRLF_AUTO
resource_file.h, 74
CONF_ITEM_TERM_ECHO
resource_file.h, 74
CONF_ITEM_TERM_FONT
resource_file.h, 74
CONF_ITEM_TERM_FOREGROUND_ALPHA
resource_file.h, 75
CONF_ITEM_TERM_FOREGROUND_BLUE
resource_file.h, 75
CONF_ITEM_TERM_FOREGROUND_GREEN
resource_file.h, 75
CONF_ITEM_TERM_FOREGROUND_RED
resource_file.h, 75
CONF_ITEM_TERM_MACROS
resource_file.h, 74
CONF_ITEM_TERM_RAW_FILENAME
resource_file.h, 74
CONF_ITEM_TERM_ROWS
resource_file.h, 74
CONF_ITEM_TERM_SCROLLBACK
resource_file.h, 74
CONF_ITEM_TERM_SHOW_CURSOR
resource_file.h, 74
CONF_ITEM_TERM_TIMESTAMP
resource_file.h, 74
CONF_ITEM_TERM_VISUAL_BELL
resource_file.h, 75
CONF_ITEM_TERM_WAIT_CHAR
resource_file.h, 74
CONF_ITEM_TERM_WAIT_DELAY
resource_file.h, 74
config
_GtkTerm, 10
config_file
GtkTermConfigurationPrivate, 20
CONFIGURATION_FILENAME
resource_file.c, 69
convert_macros_to_string
macros.c, 63
macros.h, 65
convert_string_to_macros
macros.c, 63
macros.h, 66
create_buffer
buffer.c, 34
buffer.h, 39
create_window
gtkterm_window.c, 54
gtkterm_window.h, 56
crlfauto
term_config_t, 30

DEFAULT_BAUDRATE
defaults.h, 44
DEFAULT_BITS
defaults.h, 44
DEFAULT_CHAR
defaults.h, 44
DEFAULT_DELAY
defaults.h, 45
DEFAULT_DELAY_RS485

defaults.h, 45
DEFAULT_ECHO
defaults.h, 45
default_filename
files.c, 48
files.h, 49
DEFAULT_FLOW
defaults.h, 45
DEFAULT_FONT
defaults.h, 45
DEFAULT_PARITY
defaults.h, 45
DEFAULT_PORT
defaults.h, 45
DEFAULT_SCROLLBACK
defaults.h, 46
DEFAULT_SECTION
resource_file.h, 73
DEFAULT_STOPBITS
defaults.h, 46
DEFAULT_VISUAL_BELL
defaults.h, 46
defaults.h, 43, 47
BUFFER_LENGTH, 44
DEFAULT_BAUDRATE, 44
DEFAULT_BITS, 44
DEFAULT_CHAR, 44
DEFAULT_DELAY, 45
DEFAULT_DELAY_RS485, 45
DEFAULT_ECHO, 45
DEFAULT_FLOW, 45
DEFAULT_FONT, 45
DEFAULT_PARITY, 45
DEFAULT_PORT, 45
DEFAULT_SCROLLBACK, 46
DEFAULT_STOPBITS, 46
DEFAULT_VISUAL_BELL, 46
LINE_FEED, 46
MAX_SECTION_LENGTH, 46
POLL_DELAY, 46
RECEIVE_BUFFER, 46
TRANSMIT_BUFFER, 47
delay
term_config_t, 30
delete_buffer
buffer.c, 35
buffer.h, 39
disable_port_lock
port_config_t, 28
display
interface.h, 61

echo
term_config_t, 30

files.c, 47
default_filename, 48
files.h, 48, 49
add_input, 48
default_filename, 49
save_raw_file, 48
send_raw_file, 48
waiting_for_char, 49
flow_control
port_config_t, 28
font
term_config_t, 30
foreground_color
term_config_t, 31
fullscreen
_GtkTermWindow, 17

g_config_group
_GtkTerm, 10
g_port_group
_GtkTerm, 10
g_term_group
_GtkTerm, 11
cmdline.h, 43
get_shortcuts
macros.c, 63
macros.h, 66
GtkTerm
gtkterm.h, 52
gtkterm.c, 50
gtkterm_signals, 50
main, 50
gtkterm.h, 51, 53
GtkTerm, 52
gtkterm_signals, 53
GTKTERM_TYPE_APP, 52
LAST_GTKTERM_SIGNAL, 53
SIGNAL_GTKTERM_CONFIG_SERIAL, 53
SIGNAL_GTKTERM_CONFIG_TERMINAL, 53
SIGNAL_GTKTERM_COPY_SECTION, 53
SIGNAL_GTKTERM_LOAD_CONFIG, 53
SIGNAL_GTKTERM_PRINT_SECTION, 53
SIGNAL_GTKTERM_REMOVE_SECTION, 53
SIGNAL_GTKTERM_SAVE_CONFIG, 53
SIGNAL_GTKTERM_TERMINAL_CHANGED, 53
gtkterm_add_cmdline_options
cmdline.c, 42
cmdline.h, 42
gtkterm_configuration_default_configuration
resource_file.c, 69
gtkterm_configuration_new
resource_file.h, 75
gtkterm_configuration_validate
resource_file.c, 70
gtkterm_serial_port_get_string
serial.c, 78
serial.h, 80
gtkterm_serial_port_new
serial.c, 78
serial.h, 80
gtkterm_serial_port_status
serial.c, 78
serial.h, 80

gtkterm_signals
    gtkterm.c, 50
    gtkterm.h, 53
gtkterm_terminal_new
    terminal.c, 82
    terminal.h, 84
GTKTERM_TYPE_APP
    gtkterm.h, 52
GTKTERM_TYPE_CONFIGURATION
    resource_file.h, 73
GTKTERM_TYPE_GTKTERM_WINDOW
    gtkterm_window.h, 56
GTKTERM_TYPE_SERIAL_PORT
    serial.h, 79
GTKTERM_TYPE_TERMINAL
    terminal.h, 84
gtkterm_window.c, 54
    create_window, 54
    set_window_title, 55
gtkterm_window.h, 55, 57
    create_window, 56
    GTKTERM_TYPE_GTKTERM_WINDOW, 56
    GtkTermWindow, 56
GtkTermConfiguration
    resource_file.h, 74
GtkTermConfigurationItems
    resource_file.c, 71
    resource_file.h, 76
GtkTermConfigurationPrivate, 20
    config_file, 20
    key_file, 21
GtkTermSerialPort
    serial.h, 79
GtkTermSerialPortPrivate, 21
    port_conf, 22
    serial_port_fd, 22
    termios_save, 22
GtkTermTerminalPrivate, 23
    app, 23
    macros, 24
    main_window, 24
    port_conf, 24
    section, 24
    serial_port, 24
    term_conf, 24
    view_mode, 25
GtkTermWindow
    gtkterm_window.h, 56

height
    _GtkTermWindow, 17
HEXADECIMAL_VIEW
    interface.h, 60

infobar
    _GtkTermWindow, 17
insert_timestamp
    buffer.c, 35
interface.c, 57
    show_message, 58
    timestamp_on, 58
    virt_col_pos, 59
interface.h, 59, 61
    ASCII_VIEW, 59
    display, 61
    HEXADECIMAL_VIEW, 60
    MSG_ERR, 60
    MSG_WRN, 60
    show_message, 60
    Text, 61

key_file
    GtkTermConfigurationPrivate, 21

LAST_GTKTERM_SIGNAL
    gtkterm.h, 53
LINE_FEED
    defaults.h, 46

macro_count
    macros.c, 63
    macros.h, 66
macro_t, 25
    action, 26
    closure, 26
    shortcut, 26
macros
    GtkTermTerminalPrivate, 24
    macros.c, 64
    macros.h, 67
macros.c, 61
    COLUMN_ACTION, 62
    COLUMN_SHORTCUT, 62
    convert_macros_to_string, 63
    convert_string_to_macros, 63
    get_shortcuts, 63
    macro_count, 63
    macros, 64
    nr_of_macros, 64
    NUM_COLUMNS, 62
    remove_shortcuts, 64
macros.h, 65, 67
    add_shortcuts, 65
    convert_macros_to_string, 65
    convert_string_to_macros, 66
    get_shortcuts, 66
    macro_count, 66
    macros, 67
    remove_shortcuts, 66
main
    gtkterm.c, 50
main_window
    GtkTermTerminalPrivate, 24
MAX_SECTION_LENGTH
    buffer.c, 34
    defaults.h, 46
maximized
    _GtkTermWindow, 17

menubutton
_GtkTermWindow, 18
message
_GtkTermWindow, 18
MSG_ERR
interface.h, 60
MSG_WRN
interface.h, 60

N_PROPS
serial.c, 78
terminal.c, 82
nr_of_macros
macros.c, 64
NUM_COLUMNS
macros.c, 62

on_set_config_options
resource_file.c, 71
resource_file.h, 75
overlapped
buffer.c, 37

parent_class
_GtkTermConfigurationClass, 12
_GtkTermSerialPortClass, 14
parent_instance
_GtkTerm, 11
_GtkTermConfiguration, 12
_GtkTermSerialPort, 13
_GtkTermWindow, 18
parity
port_config_t, 28
POLL_DELAY
defaults.h, 46
port
port_config_t, 28
port_conf
GtkTermSerialPortPrivate, 22
GtkTermTerminalPrivate, 24
port_config_t, 27
baudrate, 27
bits, 27
disable_port_lock, 28
flow_control, 28
parity, 28
port, 28
rs485_rts_time_after_transmit, 28
rs485_rts_time_before_transmit, 28
stopbits, 28
PROP_0
serial.c, 78
terminal.c, 82
PROP_GTKTERM_APP
terminal.c, 82
PROP_MAIN_WINDOW
terminal.c, 82
PROP_PORT_CONFIG
serial.c, 78
PROP_SECTION
terminal.c, 82
put_chars
buffer.c, 35
buffer.h, 39

README_source.md, 33
RECEIVE_BUFFER
defaults.h, 46
remove_shortcuts
macros.c, 64
macros.h, 66
resource_file.c, 68
BUFFER_LENGTH, 69
check_keyfile, 69
CONFIGURATION_FILENAME, 69
gtkterm_configuration_default_configuration, 69
gtkterm_configuration_validate, 70
GtkTermConfigurationItems, 71
on_set_config_options, 71
resource_file.h, 72, 76
CONF_ITEM_LAST, 75
CONF_ITEM_LENGTH, 73
CONF_ITEM_SERIAL_BAUDRATE, 74
CONF_ITEM_SERIAL_BITS, 74
CONF_ITEM_SERIAL_DISABLE_PORT_LOCK, 74
CONF_ITEM_SERIAL_FLOW_CONTROL, 74
CONF_ITEM_SERIAL_PARITY, 74
CONF_ITEM_SERIAL_PORT, 74
CONF_ITEM_SERIAL_RS485_RTS_TIME_AFTER_TX, 74
CONF_ITEM_SERIAL_RS485_RTS_TIME_BEFORE_TX, 74
CONF_ITEM_SERIAL_STOPBITS, 74
CONF_ITEM_TERM_BACKGROUND_ALPHA, 75
CONF_ITEM_TERM_BACKGROUND_BLUE, 75
CONF_ITEM_TERM_BACKGROUND_GREEN, 75
CONF_ITEM_TERM_BACKGROUND_RED, 75
CONF_ITEM_TERM_BLOCK_CURSOR, 74
CONF_ITEM_TERM_COLS, 74
CONF_ITEM_TERM_CRLF_AUTO, 74
CONF_ITEM_TERM_ECHO, 74
CONF_ITEM_TERM_FONT, 74
CONF_ITEM_TERM_FOREGROUND_ALPHA, 75
CONF_ITEM_TERM_FOREGROUND_BLUE, 75
CONF_ITEM_TERM_FOREGROUND_GREEN, 75
CONF_ITEM_TERM_FOREGROUND_RED, 75
CONF_ITEM_TERM_MACROS, 74
CONF_ITEM_TERM_RAW_FILENAME, 74
CONF_ITEM_TERM_ROWS, 74
CONF_ITEM_TERM_SCROLLBACK, 74
CONF_ITEM_TERM_SHOW_CURSOR, 74
CONF_ITEM_TERM_TIMESTAMP, 74
CONF_ITEM_TERM_VISUAL_BELL, 75
CONF_ITEM_TERM_WAIT_CHAR, 74
CONF_ITEM_TERM_WAIT_DELAY, 74

DEFAULT_SECTION, 73
gtkterm_configuration_new, 75
GTKTERM_TYPE_CONFIGURATION, 73
GtkTermConfiguration, 74
GtkTermConfigurationItems, 76
on_set_config_options, 75
rows
term_config_t, 31
rs485_rts_time_after_transmit
port_config_t, 28
rs485_rts_time_before_transmit
port_config_t, 28

save_raw_file
files.h, 48
scrollback
term_config_t, 31
scrolled_window
_GtkTermWindow, 18
search_bar
_GtkTermWindow, 18
section
_GtkTerm, 11
GtkTermTerminalPrivate, 24
send_raw_file
files.h, 48
serial.c, 77
gtkterm_serial_port_get_string, 78
gtkterm_serial_port_new, 78
gtkterm_serial_port_status, 78
N_PROPS, 78
PROP_0, 78
PROP_PORT_CONFIG, 78
serial.h, 79, 81
gtkterm_serial_port_get_string, 80
gtkterm_serial_port_new, 80
gtkterm_serial_port_status, 80
GTKTERM_TYPE_SERIAL_PORT, 79
GtkTermSerialPort, 79
serial_port
GtkTermTerminalPrivate, 24
serial_port_fd
GtkTermSerialPortPrivate, 22
set_clear_func
buffer.c, 35
buffer.h, 39
set_display_func
buffer.c, 35
buffer.h, 39
set_window_title
gtkterm_window.c, 55
shortcut
macro_t, 26
show_cursor
term_config_t, 31
show_message
interface.c, 58
interface.h, 60
SIGNAL_GTKTERM_CONFIG_SERIAL
gtkterm.h, 53
SIGNAL_GTKTERM_CONFIG_TERMINAL
gtkterm.h, 53
SIGNAL_GTKTERM_COPY_SECTION
gtkterm.h, 53
SIGNAL_GTKTERM_LOAD_CONFIG
gtkterm.h, 53
SIGNAL_GTKTERM_PRINT_SECTION
gtkterm.h, 53
SIGNAL_GTKTERM_REMOVE_SECTION
gtkterm.h, 53
SIGNAL_GTKTERM_SAVE_CONFIG
gtkterm.h, 53
SIGNAL_GTKTERM_TERMINAL_CHANGED
gtkterm.h, 53
status_config
_GtkTermWindow, 18
status_config_message
_GtkTermWindow, 19
status_message
_GtkTermWindow, 19
status_serial_signal
_GtkTermWindow, 19
statusbox
_GtkTermWindow, 19
stopbits
port_config_t, 28

term_conf
GtkTermTerminalPrivate, 24
term_config_t, 29
background_color, 30
block_cursor, 30
char_queue, 30
columns, 30
crlfauto, 30
delay, 30
echo, 30
font, 30
foreground_color, 31
rows, 31
scrollback, 31
show_cursor, 31
timestamp, 31
visual_bell, 31
terminal.c, 81
gtkterm_terminal_new, 82
N_PROPS, 82
PROP_0, 82
PROP_GTKTERM_APP, 82
PROP_MAIN_WINDOW, 82
PROP_SECTION, 82
terminal.h, 83, 85
gtkterm_terminal_new, 84
GTKTERM_TYPE_TERMINAL, 84
terminal_window
_GtkTermWindow, 19
termios_save
GtkTermSerialPortPrivate, 22

Text
    interface.h, 61
timestamp
    term_config_t, 31
timestamp_on
    buffer.c, 37
    interface.c, 58
TIMESTAMP_SIZE
    buffer.c, 34
toolmenu
    _GtkTermWindow, 19
TRANSMIT_BUFFER
    defaults.h, 47

unset_clear_func
    buffer.c, 35
    buffer.h, 39
unset_display_func
    buffer.c, 36
    buffer.h, 40

view_mode
    GtkTermTerminalPrivate, 25
virt_col_pos
    buffer.c, 37
    interface.c, 59
visual_bell
    term_config_t, 31
vte_class
    _GtkTermTerminalClass, 15
vte_object
    _GtkTermTerminal, 14

waiting_for_char
    files.h, 49
width
    _GtkTermWindow, 20
write_buffer
    buffer.c, 36
    buffer.h, 40
write_buffer_with_func
    buffer.c, 36
    buffer.h, 40
write_func
    buffer.c, 37