GTKTerm

Generated by Doxygen 1.9.4

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: 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 base for the terminal windows. Getting configuration for the terminal windows 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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

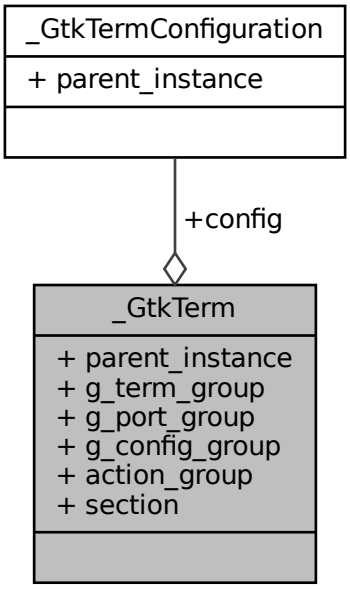
Class Documentation

4.1 _GtkTerm Struct Reference

The main GtkTerm application class.

`#include <gtkterm.h>`

Collaboration diagram for _GtkTerm:



Public Attributes

- GtkApplication parent_instance
- GOptionGroup * g_term_group
- GOptionGroup * g_port_group
- GOptionGroup * g_config_group
- GActionGroup * action_group
- GtkTermConfiguration * config

  App action group.
- char * section

  The Key file with the configurations.

4.1.1 Detailed Description

The main GtkTerm application class.

All application specific variables are defined here.

4.1.2 Member Data Documentation

4.1.2.1 action_group

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

4.1.2.2 config

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

App action group.

4.1.2.3 g_config_group

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

Referenced by gtkterm_add_cmdline_options().

4.1.2.4 g_port_group

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

Referenced by gtkterm_add_cmdline_options().

4.1.2.5 g_term_group

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

Referenced by gtkterm_add_cmdline_options().

4.1.2.6 parent_instance

```
GtkApplication _GtkTerm::parent_instance
```

4.1.2.7 section

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

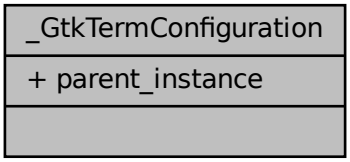
The Key file with the configurations.

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

- gtkterm.h

4.2 _GtkTermConfiguration Struct Reference

Collaboration diagram for _GtkTermConfiguration:



Public Attributes

- GObject parent_instance

4.2.1 Member Data Documentation

4.2.1.1 parent_instance

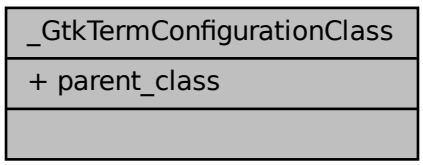
`GObject _GtkTermConfiguration::parent_instance`

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

- resource_file.c

4.3 _GtkTermConfigurationClass Struct Reference

Collaboration diagram for _GtkTermConfigurationClass:



Public Attributes

- GObjectClass parent_class

4.3.1 Member Data Documentation

4.3.1.1 parent_class

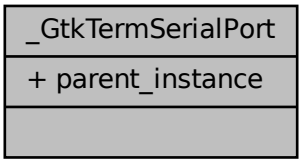
`GObjectClass _GtkTermConfigurationClass::parent_class`

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

- resource_file.c

4.4 _GtkTermSerialPort Struct Reference

Collaboration diagram for _GtkTermSerialPort:



Public Attributes

- GObject parent_instance

4.4.1 Member Data Documentation

4.4.1.1 parent_instance

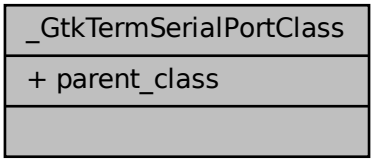
`GObject _GtkTermSerialPort::parent_instance`

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

- serial.c

4.5 _GtkTermSerialPortClass Struct Reference

Collaboration diagram for _GtkTermSerialPortClass:



Public Attributes

- GObjectClass parent_class

4.5.1 Member Data Documentation

4.5.1.1 parent_class

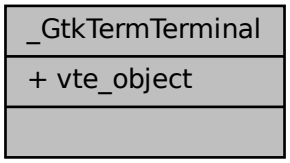
```
GObjectClass _GtkTermSerialPortClass::parent_class
```

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

- serial.c

4.6 _GtkTermTerminal Struct Reference

Collaboration diagram for _GtkTermTerminal:



Public Attributes

- VteTerminal vte_object

4.6.1 Member Data Documentation

4.6.1.1 vte_object

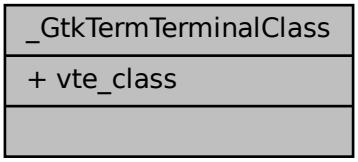
```
VteTerminal _GtkTermTerminal::vte_object
```

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

- terminal.c

4.7 _GtkTermTerminalClass Struct Reference

Collaboration diagram for _GtkTermTerminalClass:



Public Attributes

- VteTerminalClass vte_class

4.7.1 Member Data Documentation

4.7.1.1 vte_class

```
VteTerminalClass _GtkTermTerminalClass::vte_class
```

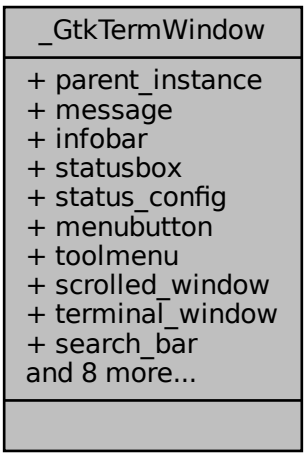
The documentation for this struct was generated from the following file:

- terminal.c

4.8 _GtkTermWindow Struct Reference

The main GtkTermWindow class.

Collaboration diagram for _GtkTermWindow:



Public Attributes

- GtkApplicationWindow parent_instance
- GtkWidget * message
- GtkWidget * infobar

  Message for the infobar.
- GtkBox * statusbox

  Infobar.
- GtkBox * status_config

  Box for statusbar messages.
- GtkWidget * menubutton

  Displays the actual used configuration.
- GMenuModel * toolmenu

  Toolbar.
- GtkScrolledWindow * scrolled_window

  Menu.
- GtkTermTerminal * terminal_window

  Make the terminal window scrolled.
- GtkWidget * search_bar

  The terminal window.
- GActionGroup * action_group

  Searchbar.
- GtkWidget * status_config_message [3]

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

4.8.1 Detailed Description

The main GtkTermWindow class.

MainWindow specific variables here.

4.8.2 Member Data Documentation

4.8.2.1 action_group

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

Searchbar.

4.8.2.2 fullscreen

```
bool _GtkTermWindow::fullscreen
```

4.8.2.3 height

```
int _GtkTermWindow::height
```

4.8.2.4 infobar

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

Message for the infobar.

4.8.2.5 maximized

```
bool _GtkTermWindow::maximized
```

4.8.2.6 menubutton

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

Displays the actual used configuration.

4.8.2.7 message

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

4.8.2.8 parent_instance

```
GtkApplicationWindow _GtkTermWindow::parent_instance
```

4.8.2.9 scrolled_window

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

Menu.

Referenced by create_window().

4.8.2.10 search_bar

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

The terminal window.

4.8.2.11 status_config

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

Box for statusbar messages.

4.8.2.12 status_config_message

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

Window action group.

4.8.2.13 status_message

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

4.8.2.14 status_serial_signal

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

4.8.2.15 statusbox

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

Infobar.

4.8.2.16 terminal_window

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

Make the terminal window scrolled.

Referenced by create_window().

4.8.2.17 toolmenu

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

Toolbar.

4.8.2.18 width

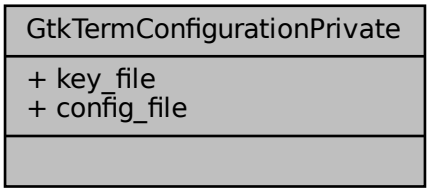
```
int _GtkTermWindow::width
```

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

- gtkterm_window.c

4.9 GtkTermConfigurationPrivate Struct Reference

Collaboration diagram for GtkTermConfigurationPrivate:



Public Attributes

- GKeyFile * key_file
- GFile * config_file

  The memory loaded keyfile.

4.9.1 Member Data Documentation

4.9.1.1 config_file

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

The memory loaded keyfile.

4.9.1.2 key_file

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

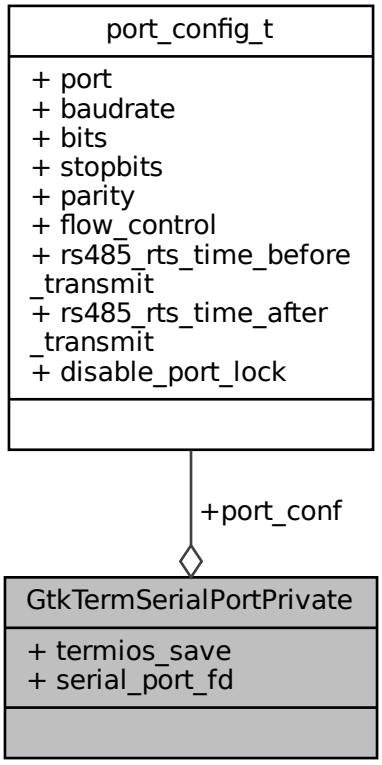
Referenced by check_keyfile(), gtkterm_configuration_default_configuration(), gtkterm_configuration_validate(), and on_set_config_options().

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

- resource_file.c

4.10 GtkTermSerialPortPrivate Struct Reference

Collaboration diagram for GtkTermSerialPortPrivate:



Public Attributes

- port_config_t * port_conf
- struct termios termios_save
- int serial_port_fd

4.10.1 Member Data Documentation

4.10.1.1 port_conf

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

Referenced by gtkterm_serial_port_get_string().

4.10.1.2 serial_port_fd

```
int GtkTermSerialPortPrivate::serial_port_fd
```

Referenced by gtkterm_serial_port_get_string(), and gtkterm_serial_port_status().

4.10.1.3 termios_save

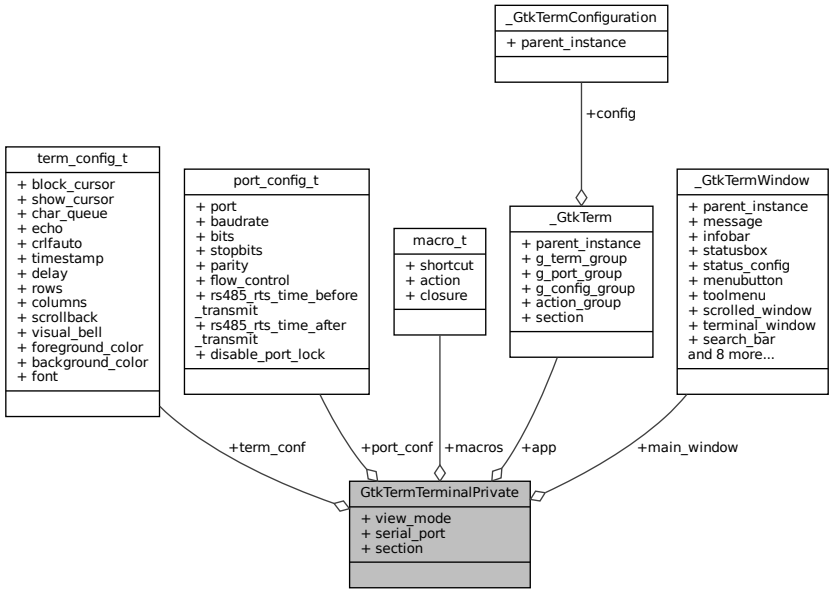
```
struct termios GtkTermSerialPortPrivate::termios_save
```

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

- serial.c

4.11 GtkTermTerminalPrivate Struct Reference

Collaboration diagram for GtkTermTerminalPrivate:



Public Attributes

- uint8_t view_mode
- GtkTermSerialPort * serial_port

  ASCII or HEX view mode.
- term_config_t * term_conf
- port_config_t * port_conf

  The configuration loaded from the keyfile.
- macro_t * macros

  Port configuration used in this terminal.
- char * section

  TODO: convert macros -> object.
- GtkTerm * app

  Section used in this terminal for configuration from config file.
- GtkTermWindow * main_window

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

4.11.1 Member Data Documentation

4.11.1.1 app

```
GtkTerm* GtkTermTerminalPrivate::app
```

Section used in this terminal for configuration from config file.

4.11.1.2 macros

```
macro_t* GtkTermTerminalPrivate::macros
```

Port configuration used in this terminal.

4.11.1.3 main_window

```
GtkTermWindow* GtkTermTerminalPrivate::main_window
```

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

4.11.1.4 port_conf

```
port_config_t* GtkTermTerminalPrivate::port_conf
```

The configuration loaded from the keyfile.

4.11.1.5 section

```
char* GtkTermTerminalPrivate::section
```

TODO: convert macros -> object.

4.11.1.6 serial_port

```
GtkTermSerialPort* GtkTermTerminalPrivate::serial_port
```

ASCII or HEX view mode.

4.11.1.7 term_conf

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

4.11.1.8 view_mode

```
uint8_t GtkTermTerminalPrivate::view_mode
```

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

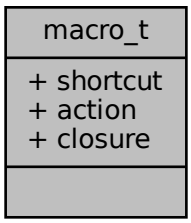
- terminal.c

4.12 macro_t Struct Reference

TODO: Migrate to GObject.

`#include <macros.h>`

Collaboration diagram for macro_t:



Public Attributes

- char * shortcut
- char * action

  Shortcut of the macro.
- GClosure * closure

  Command to perform.

4.12.1 Detailed Description

TODO: Migrate to GObject.

Define macro structure type

4.12.2 Member Data Documentation

4.12.2.1 action

`char* macro_t::action`

Shortcut of the macro.

Referenced by convert_macros_to_string(), and convert_string_to_macros().

4.12.2.2 closure

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

Command to perform.

4.12.2.3 shortcut

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

Referenced by convert_macros_to_string(), and convert_string_to_macros().

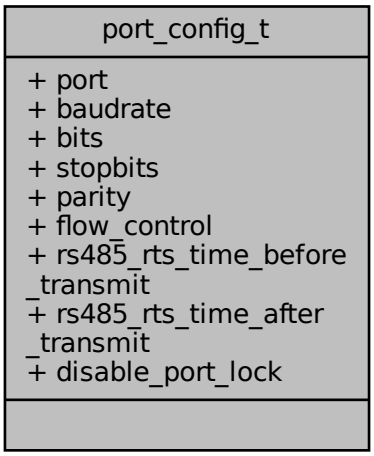
The documentation for this struct was generated from the following file:

- macros.h

4.13 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

4.13.1 Member Data Documentation

4.13.1.1 baudrate

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

Referenced by gtkterm_serial_port_get_string().

4.13.1.2 bits

```
int port_config_t::bits
```

Referenced by gtkterm_serial_port_get_string().

4.13.1.3 disable_port_lock

```
bool port_config_t::disable_port_lock
```

4.13.1.4 flow_control

```
int port_config_t::flow_control
```

4.13.1.5 parity

```
int port_config_t::parity
```

Referenced by gtkterm_serial_port_get_string().

4.13.1.6 port

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

Referenced by gtkterm_serial_port_get_string().

4.13.1.7 rs485_rts_time_after_transmit

```
int port_config_t::rs485_rts_time_after_transmit
```

4.13.1.8 rs485_rts_time_before_transmit

```
int port_config_t::rs485_rts_time_before_transmit
```

4.13.1.9 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

4.14 term_config_t Struct Reference

`#include <terminal.h>`

Collaboration diagram for term_config_t:

Public Attributes

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

4.14.1 Member Data Documentation

4.14.1.1 background_color

```
GdkRGBA term_config_t::background_color
```

4.14.1.2 block_cursor

```
bool term_config_t::block_cursor
```

4.14.1.3 char_queue

```
char term_config_t::char_queue
```

4.14.1.4 columns

```
int term_config_t::columns
```

4.14.1.5 crlfauto

```
bool term_config_t::crlfauto
```

4.14.1.6 delay

```
int term_config_t::delay
```

4.14.1.7 echo

```
bool term_config_t::echo
```

4.14.1.8 font

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

4.14.1.9 foreground_color

```
GdkRGBA term_config_t::foreground_color
```

4.14.1.10 rows

```
int term_config_t::rows
```

4.14.1.11 scrollback

```
int term_config_t::scrollback
```

4.14.1.12 show_cursor

```
bool term_config_t::show_cursor
```

4.14.1.13 timestamp

```
bool term_config_t::timestamp
```

4.14.1.14 visual_bell

```
bool term_config_t::visual_bell
```

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

- terminal.h

Chapter 5

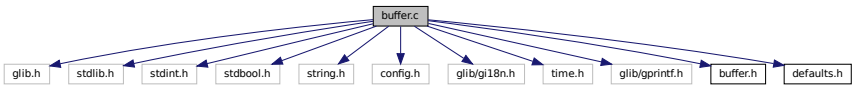
File Documentation

5.1 README_source.md File Reference

5.2 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

5.2.1 Macro Definition Documentation

5.2.1.1 MAX_SECTION_LENGTH

```
#define MAX_SECTION_LENGTH 32
```

5.2.1.2 TIMESTAMP_SIZE

```
#define TIMESTAMP_SIZE 50
```

5.2.2 Function Documentation

5.2.2.1 clear_buffer()

```
void clear_buffer (
            void  )
```

References clear_func.

5.2.2.2 create_buffer()

```
void create_buffer (
            void  )
```

5.2.2.3 delete_buffer()

```
void delete_buffer (
            void  )
```

5.2.2.4 insert_timestamp()

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

5.2.2.5 put_chars()

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

References RECEIVE_BUFFER, timestamp_on, and TIMESTAMP_SIZE.

5.2.2.6 set_clear_func()

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

References clear_func.

5.2.2.7 set_display_func()

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

References write_func.

5.2.2.8 unset_clear_func()

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

References clear_func.

5.2.2.9 unset_display_func()

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

References write_func.

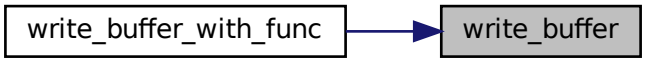
5.2.2.10 write_buffer()

```
void write_buffer (
            void  )
```

References overlapped, and write_func.

Referenced by write_buffer_with_func().

Here is the caller graph for this function:

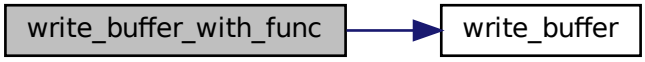


5.2.2.11 write_buffer_with_func()

```
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:



5.2.3 Variable Documentation

5.2.3.1 clear_func

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

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

5.2.3.2 overlapped

```
char overlapped
```

Referenced by write_buffer().

5.2.3.3 timestamp_on

```
bool timestamp_on  [extern]
```

Referenced by put_chars().

5.2.3.4 virt_col_pos

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

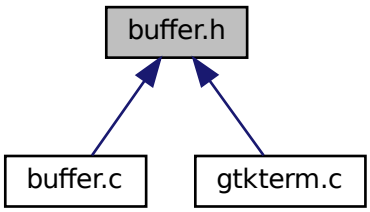
5.2.3.5 write_func

```
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().

5.3 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))

5.3.1 Macro Definition Documentation

5.3.1.1 BUFFER_SIZE

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

5.3.2 Function Documentation

5.3.2.1 clear_buffer()

```
void clear_buffer (
            void  )
```

References clear_func.

5.3.2.2 create_buffer()

```
void create_buffer (
            void  )
```

5.3.2.3 delete_buffer()

```
void delete_buffer (
            void  )
```

5.3.2.4 put_chars()

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

References RECEIVE_BUFFER, timestamp_on, and TIMESTAMP_SIZE.

5.3.2.5 set_clear_func()

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

References clear_func.

5.3.2.6 set_display_func()

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

5.3.2.7 unset_clear_func()

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

References clear_func.

5.3.2.8 unset_display_func()

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

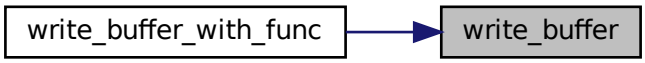
5.3.2.9 write_buffer()

```
void write_buffer (
            void  )
```

References overlapped, and write_func.

Referenced by write_buffer_with_func().

Here is the caller graph for this function:



5.3.2.10 write_buffer_with_func()

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

5.4 buffer.h

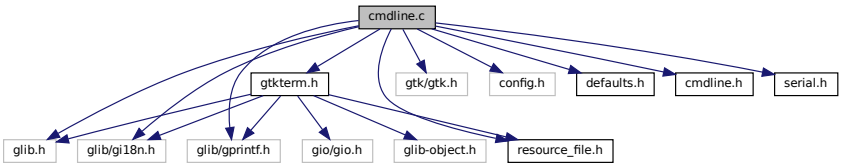
Go to the documentation of this file.

```
1 /***************************************************************************/
2 /* buffer.h                                                                */
3 /* --------                                                                */
4 /*           GTKTerm Software                                              */
5 /*                     (c) Julien Schmitt                                  */
6 /*                                                                         */
7 /* ----------------------------------------------------------------------- */
8 /*                                                                         */
9 /*   Purpose                                                               */
10 /*      Management of a local buffer of data received                     */
11 /*      - Header file -                                                    */
12 /*                                                                        */
13 /*   ChangeLog                                                            */
14 /*      - 0.99.7 :  removed auto crlf stuff - (use macros instead)        */
15 /*      - 0.98.4 :  file creation by Julien                               */
16 /*                                                                        */
17 /***************************************************************************/
18
19 #ifndef BUFFER_H_
20 #define BUFFER_H_
21
22 #define BUFFER_SIZE (128 * 1024)
23
24 void create_buffer(void);
25 void delete_buffer(void);
26 void put_chars(const char *, unsigned int, bool);
27 void clear_buffer(void);
28 void write_buffer(void);
29 void set_display_func(void (*func)(const char *, uint32_t));
30 void unset_display_func(void (*func)(const char *, uint32_t));
31 void set_clear_func(void (*func)(void));
32 void unset_clear_func(void (*func)(void));
33 void write_buffer_with_func(void (*func)(const char *, uint32_t));
34
35 #endif
```

5.5 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)

5.5.1 Function Documentation

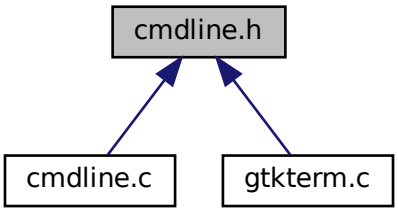
5.5.1.1 gtkterm_add_cmdline_options()

```
void gtkterm_add_cmdline_options (
            GtkTerm * app )
```

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

5.6 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

5.6.1 Function Documentation

5.6.1.1 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.

5.6.2 Variable Documentation

5.6.2.1 g_term_group

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

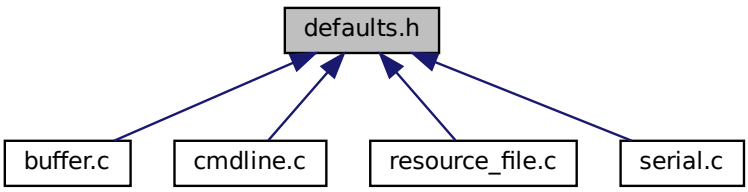
5.7 cmdline.h

Go to the documentation of this file.

```
1 /*****************************************************************************/
2 /* cmdline.h                                                                 */
3 /* ---------                                                                 */
4 /*           GTKTerm Software                                                */
5 /*                       (c) Julien Schmitt                                  */
6 /*                                                                           */
7 /* ------------------------------------------------------------------------- */
8 /*                                                                           */
9 /*   Purpose                                                                 */
10 /*      Reads the command line                                                */
11 /*      - Header file -                                                       */
12 /*                                                                           */
13 /*   ChangeLog                                                               */
14 /*      - 2.0 :  migrated to GTK4                                             */
15 /*      - 0.98 :  file creation by Julien                                     */
16 /*                                                                           */
17 /*****************************************************************************/extern GOptionGroup
     *g_term_group;
18
19 #ifndef CMDLINE_H
20 #define CMDLINE_H
21
22 void gtkterm_add_cmdline_options (GtkTerm *app);
23
24 #endif // CMDLINE_H
```

5.8 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
- #define BUFFER_LENGTH 256

  Generic defaults.
- #define MAX_SECTION_LENGTH 32

5.8.1 Macro Definition Documentation

5.8.1.1 BUFFER_LENGTH

```
#define BUFFER_LENGTH 256
```

Generic defaults.

5.8.1.2 DEFAULT_BAUDRATE

```
#define DEFAULT_BAUDRATE 115200
```

5.8.1.3 DEFAULT_BITS

```
#define DEFAULT_BITS 8
```

5.8.1.4 DEFAULT_CHAR

```
#define DEFAULT_CHAR -1
```

5.8.1.5 DEFAULT_DELAY

```
#define DEFAULT_DELAY 0
```

5.8.1.6 DEFAULT_DELAY_RS485

```
#define DEFAULT_DELAY_RS485 30
```

5.8.1.7 DEFAULT_ECHO

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

5.8.1.8 DEFAULT_FLOW

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

5.8.1.9 DEFAULT_FONT

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

Default for VTE-terminal.

5.8.1.10 DEFAULT_PARITY

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

5.8.1.11 DEFAULT_PORT

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

Default for serial ports.

5.8.1.12 DEFAULT_SCROLLBACK

```
#define DEFAULT_SCROLLBACK 10000
```

5.8.1.13 DEFAULT_STOPBITS

```
#define DEFAULT_STOPBITS 1
```

5.8.1.14 DEFAULT_VISUAL_BELL

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

5.8.1.15 LINE_FEED

```
#define LINE_FEED 0x0A
```

5.8.1.16 MAX_SECTION_LENGTH

```
#define MAX_SECTION_LENGTH 32
```

5.8.1.17 POLL_DELAY

```
#define POLL_DELAY 100
```

5.8.1.18 RECEIVE_BUFFER

```
#define RECEIVE_BUFFER 8192
```

5.8.1.19 TRANSMIT_BUFFER

```
#define TRANSMIT_BUFFER 4096
```

5.9 defaults.h

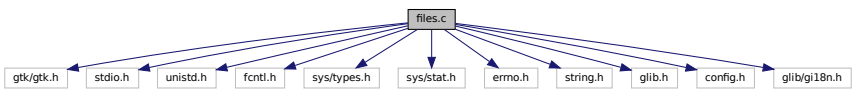
Go to the documentation of this file.

```
1 //!  Default for VTE-terminal
2 #define DEFAULT_FONT            "Monospace 12"
3 #define DEFAULT_SCROLLBACK      10000
4 #define DEFAULT_DELAY           0
5 #define DEFAULT_CHAR            -1
6 #define DEFAULT_DELAY_RS485     30
7 #define DEFAULT_ECHO            "false"
8 #define DEFAULT_VISUAL_BELL     "false"
9
10 //!  Default for serial ports
11 #define DEFAULT_PORT            "/dev/ttyS0"
12 #define DEFAULT_BAUDRATE        115200
13 #define DEFAULT_PARITY          "none"
14 #define DEFAULT_BITS            8
15 #define DEFAULT_STOPBITS        1
16 #define DEFAULT_FLOW            "none"
17
18 #define RECEIVE_BUFFER          8192
19 #define TRANSMIT_BUFFER         4096
20 #define LINE_FEED               0x0A
21 #define POLL_DELAY              100             //!  in ms (for control signals)
22
23 //!  Generic defaults
24 #define BUFFER_LENGTH           256
25 #define MAX_SECTION_LENGTH      32
```

5.10 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

5.10.1 Variable Documentation

5.10.1.1 default_filename

```
char* default_filename = NULL
```

5.11 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

5.11.1 Function Documentation

5.11.1.1 add_input()

```
void add_input (
            void  )
```

5.11.1.2 save_raw_file()

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

5.11.1.3 send_raw_file()

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

5.11.2 Variable Documentation

5.11.2.1 default_filename

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

5.11.2.2 waiting_for_char

```
gboolean waiting_for_char  [extern]
```

5.12 files.h

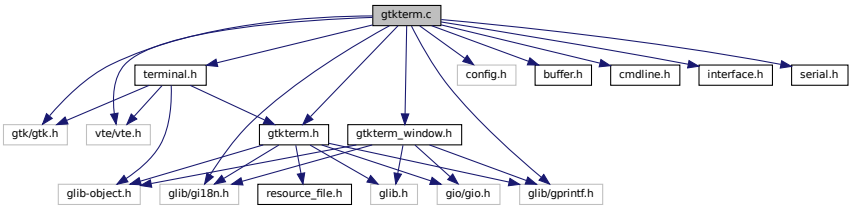
Go to the documentation of this file.

```
1 /*****************************************************************************/
2 /* files.h                                                                   */
3 /* ---------                                                                 */
4 /*           GTKTerm Software                                                */
5 /*                     (c) Julien Schmitt                                    */
6 /*                                                                           */
7 /* ------------------------------------------------------------------------- */
8 /*                                                                           */
9 /*   Purpose                                                                 */
10 /*      Raw / text file transfer management                                   */
11 /*      - Header file -                                                       */
12 /*                                                                           */
13 /*****************************************************************************/
14
15 #ifndef FILES_H_
16 #define FILES_H_
17
18 void send_raw_file (GAction *action, gpointer data);
19 void save_raw_file (GAction *action, gpointer data);
20 void add_input(void);
21
22 extern gboolean waiting_for_char;
23 extern char *default_filename;
24
25 #endif
```

5.13 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]

5.13.1 Function Documentation

5.13.1.1 main()

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

References GTKTERM_TYPE_APP.

5.13.2 Variable Documentation

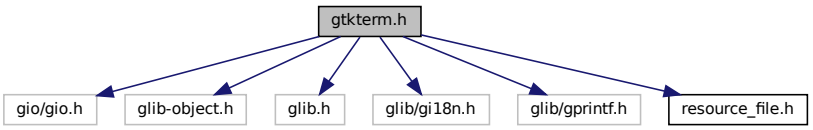
5.13.2.1 gtkterm_signals

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

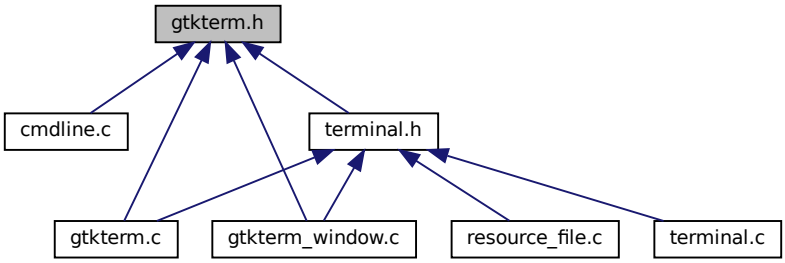
5.14 gtkterm.h File Reference

```
#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 [ ]

5.14.1 Macro Definition Documentation

5.14.1.1 GTKTERM_TYPE_APP

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

5.14.2 Typedef Documentation

5.14.2.1 GtkTerm

```
typedef struct _GtkTerm GtkTerm
```

5.14.3 Enumeration Type Documentation

5.14.3.1 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	

5.14.4 Variable Documentation

5.14.4.1 gtkterm_signals

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

5.15 gtkterm.h

Go to the documentation of this file.

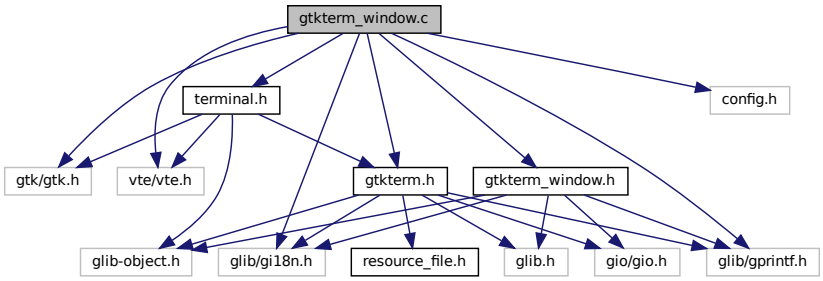
```
1
2 #ifndef GTKTERM_H
3 #define GTKTERM_H
4
5 #include <gio/gio.h>
6 #include <glib-object.h>
7 #include <glib.h>
8 #include <glib/gi18n.h>
9 #include <glib/gprintf.h>
10
11 #include "resource_file.h"
12
13 enum {
14     SIGNAL_GTKTERM_LOAD_CONFIG,
15     SIGNAL_GTKTERM_SAVE_CONFIG,
16     SIGNAL_GTKTERM_REMOVE_SECTION,
17     SIGNAL_GTKTERM_PRINT_SECTION,
18     SIGNAL_GTKTERM_COPY_SECTION,
19     SIGNAL_GTKTERM_CONFIG_TERMINAL,
20     SIGNAL_GTKTERM_CONFIG_SERIAL,
21     SIGNAL_GTKTERM_TERMINAL_CHANGED,
22     LAST_GTKTERM_SIGNAL
23 };
24
25 extern unsigned int gtkterm_signals[];
26
27 G_BEGIN_DECLS
28
29 //!  @brief The main GtkTerm application class.
30 //!  All application specific variables are defined here.
31 struct _GtkTerm {
32
33   GtkApplication parent_instance;
34
35   GOptionGroup *g_term_group;
36   GOptionGroup *g_port_group;
37   GOptionGroup *g_config_group;
38
39   GActionGroup *action_group;           //!  App action group
40
```

```
41   GtkTermConfiguration *config;         //!  The Key file with the configurations
42   char *section;                        //!  The section provided from the cli.
43 //!  Terminals have their own section pointer
44 };
45
46 #define GTKTERM_TYPE_APP gtkterm_get_type()
47 typedef struct _GtkTerm GtkTerm;
48 G_DECLARE_FINAL_TYPE (GtkTerm, gtkterm, GTKTERM, APP, GtkApplication)
49
50
51 #endif // GTKTERM_H
```

5.16 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)

5.16.1 Function Documentation

5.16.1.1 create_window()

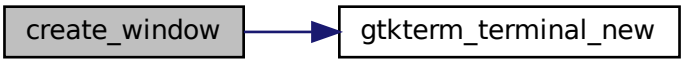
```
void create_window (
            GApplication * app )
```

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:



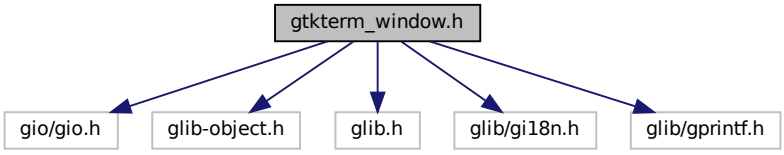
5.16.1.2 set_window_title()

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

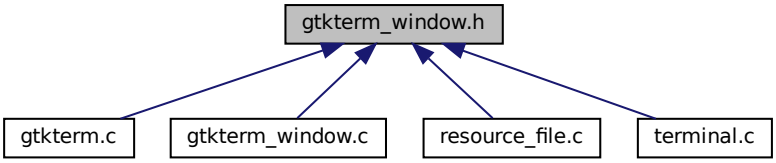
5.17 gtkterm_window.h File Reference

```
#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 *)

5.17.1 Macro Definition Documentation

5.17.1.1 GTKTERM_TYPE_GTKTERM_WINDOW

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

5.17.2 Typedef Documentation

5.17.2.1 GtkTermWindow

```
typedef struct _GtkTermWindow GtkTermWindow
```

5.17.3 Function Documentation

5.17.3.1 create_window()

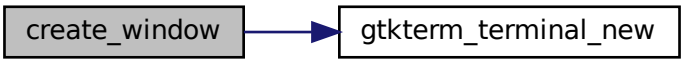
```
G_END_DECLS void create_window (
              GApplication * app )
```

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:



5.18 gtkterm_window.h

Go to the documentation of this file.

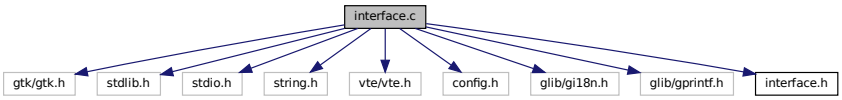
```
1 #include <gio/gio.h>
2 #include <glib-object.h>
3 #include <glib.h>
4 #include <glib/gi18n.h>
5 #include <glib/gprintf.h>
6
7 #ifndef GTKTERM_WINDOW_H
8 #define GTKTERM_WINDOW_H
9
10 G_BEGIN_DECLS
11
12 #define GTKTERM_TYPE_GTKTERM_WINDOW gtkterm_window_get_type()
13 typedef struct _GtkTermWindow GtkTermWindow;
14 G_DECLARE_FINAL_TYPE (GtkTermWindow, gtkterm_window, GTKTERM, WINDOW, GtkApplicationWindow)
15
16 G_END_DECLS
17
18 void create_window (GApplication *);
19
20 #endif // GTKTERM_WINDOW_H
```

5.19 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

5.19.1 Function Documentation

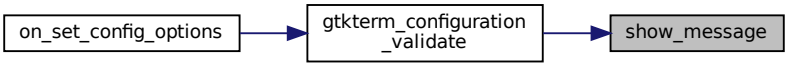
5.19.1.1 show_message()

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

References MSG_ERR.

Referenced by gtkterm_configuration_validate().

Here is the caller graph for this function:



5.19.2 Variable Documentation

5.19.2.1 timestamp_on

```
bool timestamp_on = 0
```

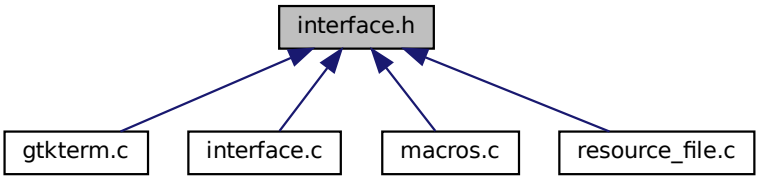
Referenced by put_chars().

5.19.2.2 virt_col_pos

```
int virt_col_pos = 0
```

5.20 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

5.20.1 Macro Definition Documentation

5.20.1.1 ASCII_VIEW

```
#define ASCII_VIEW 0
```

5.20.1.2 HEXADECIMAL_VIEW

```
#define HEXADECIMAL_VIEW 1
```

5.20.1.3 MSG_ERR

```
#define MSG_ERR 1
```

5.20.1.4 MSG_WRN

```
#define MSG_WRN 0
```

5.20.2 Function Documentation

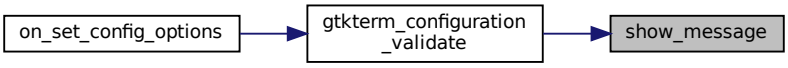
5.20.2.1 show_message()

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

References MSG_ERR.

Referenced by gtkterm_configuration_validate().

Here is the caller graph for this function:



5.20.3 Variable Documentation

5.20.3.1 display

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

5.20.3.2 Text

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

5.21 interface.h

Go to the documentation of this file.

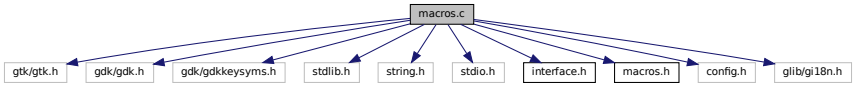
```
1 /*****************************************************************************/
2 /* interface.h                                                               */
3 /* ---------                                                                 */
4 /*           GTKTerm Software                                                */
5 /*                       (c) Julien Schmitt                                  */
6 /*                                                                           */
7 /* ------------------------------------------------------------------------- */
8 /*                                                                           */
9 /*   Purpose                                                                 */
10 /*      Functions for the management of the GUI for the main window    */
11 /*      - Header file -                                                 */
12 /*                                                                      */
13 /*****************************************************************************/
14
15 #ifndef WIDGETS_H_
16 #define WIDGETS_H_
17
18 #define MSG_WRN          0
19 #define MSG_ERR          1
20
21 #define ASCII_VIEW       0
22 #define HEXADECIMAL_VIEW 1
23
24 extern GtkWidget *Text;
25 extern GtkWidget *display;          // Serial terminal (vte)
26
27 void show_message(char *, int);
28
29 #endif
```

5.22 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 }

  TODO: Migrate to GObject.

Functions

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

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

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

Variables

- macro_t * macros = NULL
- int nr_of_macros = 0

5.22.1 Enumeration Type Documentation

5.22.1.1 anonymous enum

```
anonymous enum
```

TODO: Migrate to GObject.

Enumerator

COLUMN_SHORTCUT	
COLUMN_ACTION	
NUM_COLUMNS	

5.22.2 Function Documentation

5.22.2.1 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.

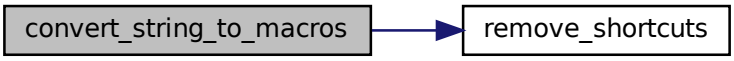
5.22.2.2 convert_string_to_macros()

```
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:



5.22.2.3 get_shortcuts()

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

References macros.

5.22.2.4 macro_count()

```
int macro_count ( )
```

References nr_of_macros.

5.22.2.5 remove_shortcuts()

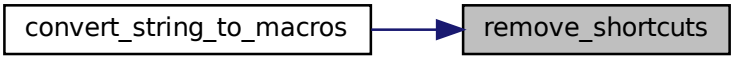
```
void remove_shortcuts (
             void  )
```

Clean up all macros

References macros.

Referenced by convert_string_to_macros().

Here is the caller graph for this function:



5.22.3 Variable Documentation

5.22.3.1 macros

```
macro_t* macros = NULL
```

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

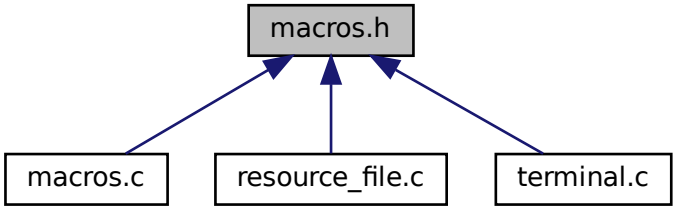
5.22.3.2 nr_of_macros

```
int nr_of_macros = 0
```

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

5.23 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)
- void add_shortcuts (void)

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

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

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

Variables

- macro_t * macros

5.23.1 Function Documentation

5.23.1.1 add_shortcuts()

```
void add_shortcuts (
            void  )
```

Remove shortcuts from accel_group and free memory.

5.23.1.2 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.

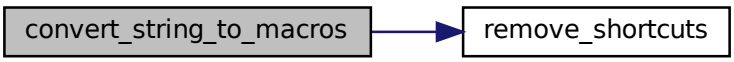
5.23.1.3 convert_string_to_macros()

```
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:



5.23.1.4 get_shortcuts()

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

5.23.1.5 macro_count()

```
int macro_count ( )
```

References nr_of_macros.

5.23.1.6 remove_shortcuts()

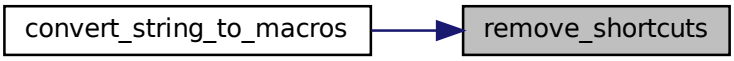
```
void remove_shortcuts (
            void  )
```

Clean up all macros

References macros.

Referenced by convert_string_to_macros().

Here is the caller graph for this function:



5.23.2 Variable Documentation

5.23.2.1 macros

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

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

5.24 macros.h

Go to the documentation of this file.

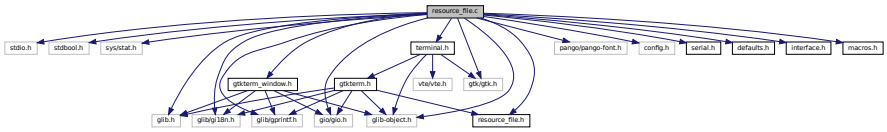
```
1 /*****************************************************************************
2 * macros.h
3 * --------
4 *           GTKTerm Software
5 *                     (c) Julien Schmitt
6 *
7 * ---------------------------------------------------------------------
8 *
9 * \brief Purpose
10 *        Functions for the management of the macros
11 *        - Header file -
12 *
13 *****************************************************************************/
14
15 #ifndef MACROS_H_
16 #define MACROS_H_
17
18 //!  TODO: Migrate to GObject
19
20 //!  Define macro structure type
21 typedef struct
22 {
23     char *shortcut;     //!  Shortcut of the macro
```

```
24     char *action;     //!  Command to perform
25     GClosure *closure;   //!
26 }
27 macro_t;
28
29 //void config_macros(GtkAction *action, gpointer data);
30 void remove_shortcuts(void);                //!  Remove shortcuts from accel_group and free memory
31 void add_shortcuts(void);                  //!
32 macro_t *get_shortcuts(gint *);             //!
33
34 void convert_string_to_macros (char **, int);
35 int convert_macros_to_string (char **);
36
37 int macro_count ();
38
39 extern macro_t *macros;
40
41 #endif
```

5.25 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

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.

5.25.1 Macro Definition Documentation

5.25.1.1 BUFFER_LENGTH

```
#define BUFFER_LENGTH 256
```

5.25.1.2 CONFIGURATION_FILENAME

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

Default configuration filename.

5.25.2 Function Documentation

5.25.2.1 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.

5.25.2.2 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.

5.25.2.3 gtkterm_configuration_validate()

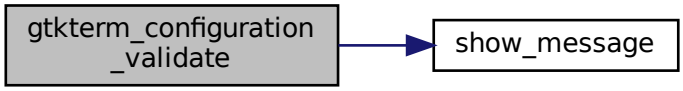
```
void gtkterm_configuration_validate (
            GtkTermConfiguration * self,
            char * section )
```

validate the configuration, given by the section

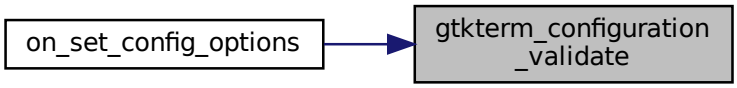
References CONF_ITEM_SERIAL_BAUDRATE, CONF_ITEM_SERIAL_BITS, CONF_ITEM_SERIAL_STOPBITS, CONF_ITEM_TERM_FONT, CONF_ITEM_TERM_WAIT_DELAY, DEFAULT_BITS, DEFAULT_DELAY, DEFAULT_FONT, DEFAULT_STOPBITS, GtkTermConfigurationItems, GtkTermConfigurationPrivate::key_file, MSG_ERR, and show_message().

Referenced by on_set_config_options().

Here is the call graph for this function:



Here is the caller graph for this function:



5.25.2.4 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)

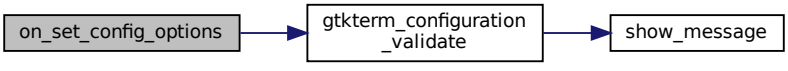
Search index for the option we want to set

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:



5.25.3 Variable Documentation

5.25.3.1 GtkTermConfigurationItems

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

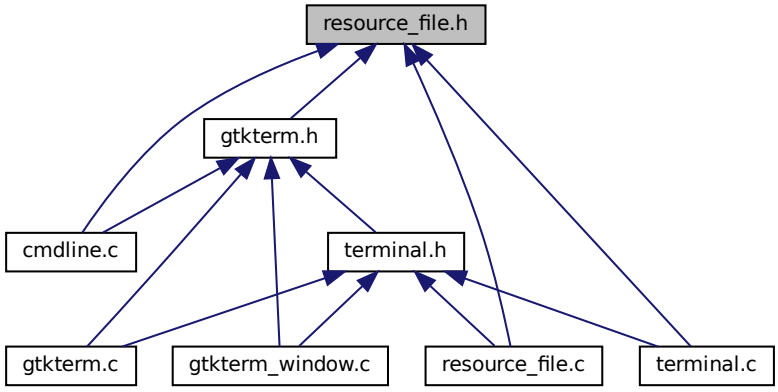
Used configuration options to hold consistency between load/save functions.

Configuration item names.

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

5.26 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"
- #define GTKTERM_TYPE_CONFIGURATION gtkterm_configuration_get_type ()

Typedefs

- typedef struct _GtkTermConfiguration GtkTermConfiguration

Enumerations

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

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

Functions

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

  Set the config option in the keyfile.

Variables

- const char GtkTermConfigurationItems [ ][CONF_ITEM_LENGTH]

  Configuration item names.

5.26.1 Macro Definition Documentation

5.26.1.1 CONF_ITEM_LENGTH

```
#define CONF_ITEM_LENGTH 32
```

5.26.1.2 DEFAULT_SECTION

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

5.26.1.3 GTKTERM_TYPE_CONFIGURATION

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

5.26.2 Typedef Documentation

5.26.2.1 GtkTermConfiguration

```
typedef struct _GtkTermConfiguration GtkTermConfiguration
```

5.26.3 Enumeration Type Documentation

5.26.3.1 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.

5.26.4 Function Documentation

5.26.4.1 gtkterm_configuration_new()

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

5.26.4.2 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)

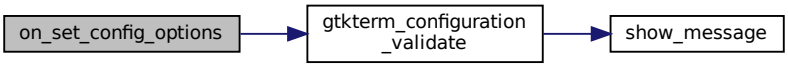
Search index for the option we want to set

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:



5.26.5 Variable Documentation

5.26.5.1 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().

5.27 resource_file.h

Go to the documentation of this file.

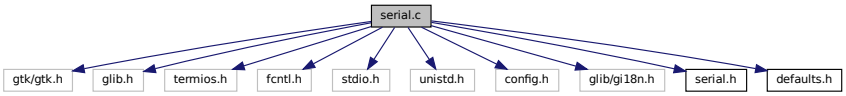
```
1 /*********************************************************************
2 * resource_file.h
3 * ---------------
4 *           GTKTerm Software
5 *                     (c) Julien Schmitt
6 *
7 * -------------------------------------------------------------------
8 *
9 *   \brief Purpose
10 *      Load and save configuration file
11 *      - Header file -
12 *
13 *********************************************************************/
14
15 #ifndef RESOURCE_FILE_H_
16 #define RESOURCE_FILE_H_
17
18 #define CONF_ITEM_LENGTH        32
19 #define DEFAULT_SECTION        "default"        //!  Default section if not specified
20
21 //!  Define all configuration items which are used
22 //!  in the resource file.  it is an index to ConfigurationItem.
23 enum {
24         CONF_ITEM_SERIAL_PORT,
25         CONF_ITEM_SERIAL_BAUDRATE,
26         CONF_ITEM_SERIAL_BITS,
27         CONF_ITEM_SERIAL_STOPBITS,
28         CONF_ITEM_SERIAL_PARITY,
29         CONF_ITEM_SERIAL_FLOW_CONTROL,
30         CONF_ITEM_TERM_WAIT_DELAY,
31         CONF_ITEM_TERM_WAIT_CHAR,
32         CONF_ITEM_SERIAL_RS485_RTS_TIME_BEFORE_TX,
33         CONF_ITEM_SERIAL_RS485_RTS_TIME_AFTER_TX,
34         CONF_ITEM_TERM_MACROS,
35         CONF_ITEM_TERM_RAW_FILENAME,
36         CONF_ITEM_TERM_ECHO,
37         CONF_ITEM_TERM_CRLF_AUTO,
38         CONF_ITEM_SERIAL_DISABLE_PORT_LOCK,
39         CONF_ITEM_TERM_FONT,
40         CONF_ITEM_TERM_TIMESTAMP,
41         CONF_ITEM_TERM_BLOCK_CURSOR,
42         CONF_ITEM_TERM_SHOW_CURSOR,
43         CONF_ITEM_TERM_ROWS,
44         CONF_ITEM_TERM_COLS,
45         CONF_ITEM_TERM_SCROLLBACK,
46         CONF_ITEM_TERM_VISUAL_BELL,
47         CONF_ITEM_TERM_FOREGROUND_RED,
48         CONF_ITEM_TERM_FOREGROUND_GREEN,
49         CONF_ITEM_TERM_FOREGROUND_BLUE,
50         CONF_ITEM_TERM_FOREGROUND_ALPHA,
51         CONF_ITEM_TERM_BACKGROUND_RED,
52         CONF_ITEM_TERM_BACKGROUND_GREEN,
53         CONF_ITEM_TERM_BACKGROUND_BLUE,
54         CONF_ITEM_TERM_BACKGROUND_ALPHA,
55         CONF_ITEM_LAST                        //!  Checking as last item in the list.
```

```
56 };
57
58 //!Configuration item names.
59 extern const char GtkTermConfigurationItems [][CONF_ITEM_LENGTH];
60
61 G_BEGIN_DECLS
62
63 #define GTKTERM_TYPE_CONFIGURATION gtkterm_configuration_get_type ()
64 G_DECLARE_FINAL_TYPE (GtkTermConfiguration, gtkterm_configuration, GTKTERM, CONFIGURATION, GObject)
65 typedef struct _GtkTermConfiguration GtkTermConfiguration;
66
67 GtkTermConfiguration *gtkterm_configuration_new (void);
68
69 bool on_set_config_options (const char *, const char *, gpointer,  GError **);
70
71 G_END_DECLS
72
73 #endif
```

5.28 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)

5.28.1 Enumeration Type Documentation

5.28.1.1 anonymous enum

```
anonymous enum
```

Enumerator

PROP_0	
PROP_PORT_CONFIG	
N_PROPS	

5.28.2 Function Documentation

5.28.2.1 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.

5.28.2.2 gtkterm_serial_port_new()

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

References GTKTERM_TYPE_SERIAL_PORT.

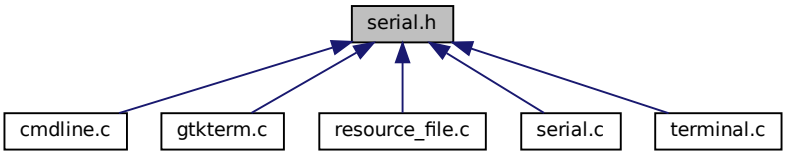
5.28.2.3 gtkterm_serial_port_status()

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

References GtkTermSerialPortPrivate::serial_port_fd.

5.29 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 *)

5.29.1 Macro Definition Documentation

5.29.1.1 GTKTERM_TYPE_SERIAL_PORT

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

5.29.2 Typedef Documentation

5.29.2.1 GtkTermSerialPort

```
typedef typedefG_BEGIN_DECLS struct _GtkTermSerialPort GtkTermSerialPort
```

5.29.3 Function Documentation

5.29.3.1 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.

5.29.3.2 gtkterm_serial_port_new()

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

References GTKTERM_TYPE_SERIAL_PORT.

5.29.3.3 gtkterm_serial_port_status()

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

References GtkTermSerialPortPrivate::serial_port_fd.

5.30 serial.h

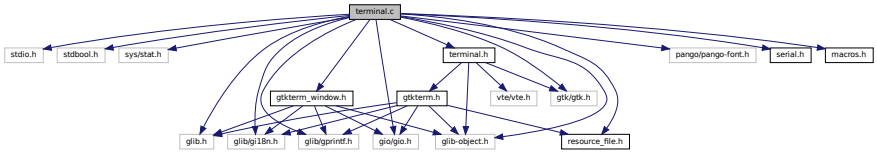
Go to the documentation of this file.

```
1 /*****************************************************************************/
2 /* serial.h                                                                  */
3 /* -------                                                                   */
4 /*           GTKTerm Software                                                */
5 /*                       (c) Julien Schmitt                                  */
6 /*                                                                           */
7 /* ------------------------------------------------------------------------- */
8 /*                                                                           */
9 /*   Purpose                                                                 */
10 /*      Serial port access functions                                          */
11 /*      - Header file -                                                       */
12 /*                                                                           */
13 /*****************************************************************************/
14
15 #ifndef SERIAL_H_
16 #define SERIAL_H_
17
18 typedef struct
19 {
20     char *port;
21     long int baudrate;              // 300 - 600 - 1200 - ...  - 2000000
22     int bits;                       // 5 - 6 - 7 - 8
23     int stopbits;                   // 1 - 2
24     int parity;                   // 0 :  None, 1 :  Odd, 2 :  Even
25     int flow_control;               // 0 :  None, 1 :  Xon/Xoff, 2 :  RTS/CTS, 3 :  RS485halfduplex
26     int rs485_rts_time_before_transmit;
27     int rs485_rts_time_after_transmit;
28     bool disable_port_lock;
29
30 } port_config_t;
31
32 G_BEGIN_DECLS
33
34 typedef struct _GtkTermSerialPort GtkTermSerialPort;
35
36 #define GTKTERM_TYPE_SERIAL_PORT gtkterm_serial_port_get_type ()
37 G_DECLARE_FINAL_TYPE (GtkTermSerialPort, gtkterm_serial_port, GTKTERM, SERIAL_PORT, GObject)
38
39 GtkTermSerialPort *gtkterm_serial_port_new (port_config_t *);
40
41 G_END_DECLS
42
43 char* gtkterm_serial_port_get_string (GtkTermSerialPort *);
44 int gtkterm_serial_port_status (GtkTermSerialPort *);
45
46 #endif
```

5.31 terminal.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 "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)

5.31.1 Enumeration Type Documentation

5.31.1.1 anonymous enum

```
anonymous enum
```

Enumerator

PROP_0	
PROP_SECTION	
PROP_GTKTERM_APP	
PROP_MAIN_WINDOW	
N_PROPS	

5.31.2 Function Documentation

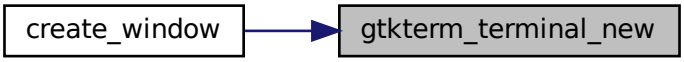
5.31.2.1 gtkterm_terminal_new()

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

References GTKTERM_TYPE_TERMINAL.

Referenced by create_window().

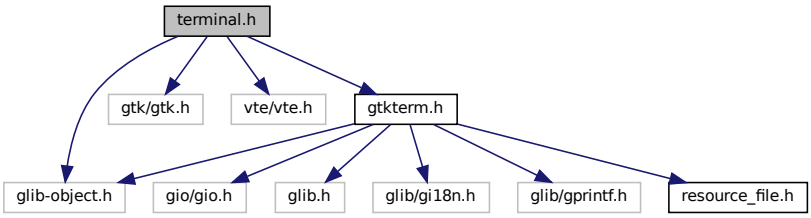
Here is the caller graph for this function:



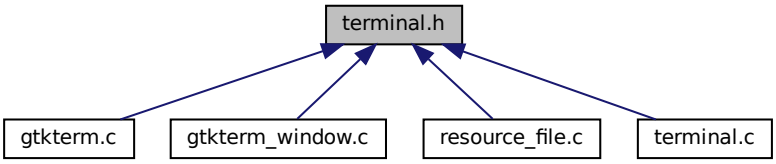
5.32 terminal.h File Reference

```
#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 *)

5.32.1 Macro Definition Documentation

5.32.1.1 GTKTERM_TYPE_TERMINAL

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

5.32.2 Function Documentation

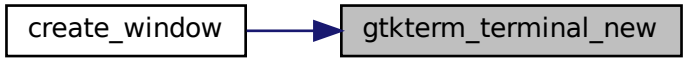
5.32.2.1 gtkterm_terminal_new()

```
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:



5.33 terminal.h

Go to the documentation of this file.

```
1 /*****************************************************************************/
2 /* terminal.h                                                                */
3 /* --------                                                                  */
4 /*           GTKTerm Software                                                */
5 /*                       (c) Julien Schmitt                                  */
6 /*                                                                           */
7 /* ------------------------------------------------------------------------- */
8 /*                                                                           */
9 /*   Purpose                                                                 */
10 /*      Handles all VTE in/output to/from serial port                         */
11 /*      - Header file -                                                       */
12 /*                                                                           */
13 /*****************************************************************************/
14 #ifndef TERMINAL_H
15 #define TERMINAL_H
16
17 #include <glib-object.h>
18 #include <gtk/gtk.h>
19 #include <vte/vte.h>
20
21 #include "gtkterm.h"
22
23 typedef struct
24 {
25     bool block_cursor;
26     bool show_cursor;
27     char char_queue;            // character in queue
28     bool echo;                  // echo local
29     bool crlfauto;              // line feed auto
30     bool timestamp;
31     int delay;                  // end of char delay:  in ms
32     int rows;
33     int columns;
34     int scrollback;
35     bool visual_bell;
36     GdkRGBA foreground_color;
37     GdkRGBA background_color;
38     PangoFontDescription *font;
39
40 } term_config_t;
41
42 G_BEGIN_DECLS
43
44 #define GTKTERM_TYPE_TERMINAL gtkterm_terminal_get_type()
45 G_DECLARE_FINAL_TYPE (GtkTermTerminal, gtkterm_terminal, GTKTERM, TERMINAL, VteTerminal)
46
47 GtkTermTerminal *gtkterm_terminal_new (char *, GtkTerm *, GtkTermWindow *);
48
49 G_END_DECLS
50
51 #endif // TERMINAL_H
```

Index