

# VimTogether

## Pre-Release

Generated by Doxygen 1.11.0



<b>1 VimTogether</b>	<b>1</b>
1.1 Table of Contents	1
1.2 1. Project Problem and Solution Statement	1
1.3 2. High Level Design Overview	1
1.3.1 Functional and Nonfunctional Requirements	1
1.3.2 System Architecture	1
1.3.3 Key Components of the Kilo	1
1.4 3. Documentation	1
<b>2 Data Structure Index</b>	<b>3</b>
2.1 Data Structures	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Data Structure Documentation</b>	<b>7</b>
4.1 abuf Struct Reference	7
4.1.1 Field Documentation	7
4.1.1.1 b	7
4.1.1.2 len	7
4.2 editorConfig Struct Reference	7
4.2.1 Field Documentation	8
4.2.1.1 coloff	8
4.2.1.2 cx	8
4.2.1.3 cy	8
4.2.1.4 dirty	8
4.2.1.5 filename	8
4.2.1.6 numrows	8
4.2.1.7 orig_termios	8
4.2.1.8 row	8
4.2.1.9 rowoff	8
4.2.1.10 rx	8
4.2.1.11 screencols	9
4.2.1.12 screenrows	9
4.2.1.13 statusmsg	9
4.2.1.14 statusmsg_time	9
4.3 erow Struct Reference	9
4.3.1 Field Documentation	9
4.3.1.1 chars	9
4.3.1.2 render	9
4.3.1.3 rsize	9
4.3.1.4 size	9
<b>5 File Documentation</b>	<b>11</b>

5.1 kilo.c File Reference	11
5.1.1 Detailed Description	13
5.1.2 Macro Definition Documentation	13
5.1.2.1 _BSD_SOURCE	13
5.1.2.2 _DEFAULT_SOURCE	13
5.1.2.3 _GNU_SOURCE	13
5.1.2.4 ABUF_INIT	13
5.1.2.5 CTRL_KEY	13
5.1.2.6 KILO_QUIT_TIMES	13
5.1.2.7 KILO_TAB_STOP	13
5.1.2.8 KILO_VERSION	14
5.1.3 Typedef Documentation	14
5.1.3.1 erow	14
5.1.4 Enumeration Type Documentation	14
5.1.4.1 editorKey	14
5.1.5 Function Documentation	14
5.1.5.1 abAppend()	14
5.1.5.2 abFree()	14
5.1.5.3 die()	14
5.1.5.4 disableRawMode()	15
5.1.5.5 editorDelChar()	15
5.1.5.6 editorDelRow()	15
5.1.5.7 editorDrawMessageBar()	15
5.1.5.8 editorDrawRows()	15
5.1.5.9 editorDrawStatusBar()	15
5.1.5.10 editorFreeRow()	15
5.1.5.11 editorInsertChar()	15
5.1.5.12 editorInsertNewline()	16
5.1.5.13 editorInsertRow()	16
5.1.5.14 editorMoveCursor()	16
5.1.5.15 editorOpen()	16
5.1.5.16 editorProcessKeypress()	16
5.1.5.17 editorPrompt()	16
5.1.5.18 editorReadKey()	16
5.1.5.19 editorRefreshScreen()	16
5.1.5.20 editorRowAppendString()	17
5.1.5.21 editorRowCxToRx()	17
5.1.5.22 editorRowDelChar()	17
5.1.5.23 editorRowInsertChar()	17
5.1.5.24 editorRowsToString()	17
5.1.5.25 editorSave()	17
5.1.5.26 editorScroll()	17

---

5.1.5.27 editorSetStatusMessage()	17
5.1.5.28 editorUpdateRow()	18
5.1.5.29 enableRawMode()	18
5.1.5.30 getCursorPosition()	18
5.1.5.31 getWindowSize()	18
5.1.5.32 initEditor()	18
5.1.5.33 main()	18
5.1.6 Variable Documentation	18
5.1.6.1 E	18
5.2 README.md File Reference	18



# Chapter 1

## VimTogether

A lightweight, terminal based task editor with real time cloud-based collaboration

### 1.1 Table of Contents

- Project Statement
- High Level Design Document
- Documentation

### 1.2 1. Project Problem and Solution Statement

Vim is a dated yet still beloved text editor known for its quick editing speeds, convenient (though complicated keyboard shortcuts), and simplicity.

VimTogether aims to replicate Vim's original functionality and improve upon the base model by simplifying complex keyboard shortcuts and adding advanced features such as live collaboration via the cloud, similar to a Google Doc.

### 1.3 2. High Level Design Overview

- Requirements
- System Architecture
- Key Components

#### 1.3.1 Functional and Nonfunctional Requirements

#### 1.3.2 System Architecture

- client/server side components
- backend database

#### 1.3.3 Key Components of the Kilo

### 1.4 3. Documentation





## Chapter 2

# Data Structure Index

### 2.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">abuf</a>	7
<a href="#">editorConfig</a>	7
<a href="#">erow</a>	9



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">kilo.c</a>	The file responsible for running the text editor . . . . .	11
------------------------	--	----



## Chapter 4

# Data Structure Documentation

### 4.1 abuf Struct Reference

#### Data Fields

- char \* [b](#)
- int [len](#)

#### 4.1.1 Field Documentation

##### 4.1.1.1 [b](#)

char\* [b](#)

##### 4.1.1.2 [len](#)

int [len](#)

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

- [kilo.c](#)

### 4.2 editorConfig Struct Reference

#### Data Fields

- int [cx](#)
- int [cy](#)
- int [rx](#)
- int [rowoff](#)
- int [coloff](#)
- int [screenrows](#)
- int [screencols](#)
- int [numrows](#)
- [erow](#) \* [row](#)
- int [dirty](#)
- char \* [filename](#)
- char [statusmsg](#) [80]
- time\_t [statusmsg\\_time](#)
- struct termios [orig\\_termios](#)

## 4.2.1 Field Documentation

### 4.2.1.1 coloff

```
int coloff
```

### 4.2.1.2 cx

```
int cx
```

### 4.2.1.3 cy

```
int cy
```

### 4.2.1.4 dirty

```
int dirty
```

### 4.2.1.5 filename

```
char* filename
```

### 4.2.1.6 numrows

```
int numrows
```

### 4.2.1.7 orig\_termios

```
struct termios orig_termios
```

### 4.2.1.8 row

```
erow* row
```

### 4.2.1.9 rowoff

```
int rowoff
```

### 4.2.1.10 rx

```
int rx
```

#### 4.2.1.11 screencols

```
int screencols
```

#### 4.2.1.12 screenrows

```
int screenrows
```

#### 4.2.1.13 statusmsg

```
char statusmsg[80]
```

#### 4.2.1.14 statusmsg\_time

```
time_t statusmsg_time
```

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

- [kilo.c](#)

## 4.3 erow Struct Reference

### Data Fields

- int [size](#)
- int [rsize](#)
- char \* [chars](#)
- char \* [render](#)

### 4.3.1 Field Documentation

#### 4.3.1.1 chars

```
char* chars
```

#### 4.3.1.2 render

```
char* render
```

#### 4.3.1.3 rsize

```
int rsize
```

#### 4.3.1.4 size

```
int size
```

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

- [kilo.c](#)





# Chapter 5

## File Documentation

### 5.1 kilo.c File Reference

The file responsible for running the text editor.

```
#include <ctype.h>
#include <errno.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdarg.h>
#include <stdlib.h>
#include <string.h>
#include <sys/ioctl.h>
#include <sys/types.h>
#include <termios.h>
#include <time.h>
#include <unistd.h>
```

#### Data Structures

- struct [erow](#)
- struct [editorConfig](#)
- struct [abuf](#)

#### Macros

- #define [\\_DEFAULT\\_SOURCE](#)
- #define [\\_BSD\\_SOURCE](#)
- #define [\\_GNU\\_SOURCE](#)
- #define [KILO\\_VERSION](#) "0.0.1"
- #define [KILO\\_TAB\\_STOP](#) 8
- #define [KILO\\_QUIT\\_TIMES](#) 3
- #define [CTRL\\_KEY\(k\)](#) ((k) & 0x1f)
- #define [ABUF\\_INIT](#) {NULL, 0}

## Typedefs

- typedef struct erow [erow](#)

## Enumerations

- enum [editorKey](#) {  
[BACKSPACE](#) = 127 , [ARROW\\_LEFT](#) = 1000 , [ARROW\\_RIGHT](#) , [ARROW\\_UP](#) ,  
[ARROW\\_DOWN](#) , [DEL\\_KEY](#) , [HOME\\_KEY](#) , [END\\_KEY](#) ,  
[PAGE\\_UP](#) , [PAGE\\_DOWN](#) }

## Functions

- void [editorSetStatusMessage](#) (const char \*fmt,...)
- void [editorRefreshScreen](#) (void)
- char \* [editorPrompt](#) (char \*prompt)
- void [die](#) (const char \*s)  
*A panic function that is called when an unforeseen error occurs.*
- void [disableRawMode](#) (void)
- void [enableRawMode](#) (void)
- int [editorReadKey](#) (void)
- int [getCursorPosition](#) (int \*rows, int \*cols)
- int [getWindowSize](#) (int \*rows, int \*cols)
- int [editorRowCxToRx](#) ([erow](#) \*row, int cx)
- void [editorUpdateRow](#) ([erow](#) \*row)
- void [editorInsertRow](#) (int at, char \*s, size\_t len)
- void [editorFreeRow](#) ([erow](#) \*row)
- void [editorDelRow](#) (int at)
- void [editorRowInsertChar](#) ([erow](#) \*row, int at, int c)
- void [editorRowAppendString](#) ([erow](#) \*row, char \*s, size\_t len)
- void [editorRowDelChar](#) ([erow](#) \*row, int at)
- void [editorInsertChar](#) (int c)
- void [editorDelChar](#) (void)
- void [editorInsertNewline](#) (void)
- char \* [editorRowsToString](#) (int \*buflen)
- void [editorOpen](#) (char \*filename)
- void [editorSave](#) (void)
- void [abAppend](#) (struct [abuf](#) \*ab, const char \*s, int len)
- void [abFree](#) (struct [abuf](#) \*ab)
- void [editorScroll](#) (void)
- void [editorDrawRows](#) (struct [abuf](#) \*ab)
- void [editorDrawStatusBar](#) (struct [abuf](#) \*ab)
- void [editorDrawMessageBar](#) (struct [abuf](#) \*ab)
- void [editorMoveCursor](#) (int key)
- void [editorProcessKeypress](#) (void)
- void [initEditor](#) (void)
- int [main](#) (int argc, char \*argv[ ])

## Variables

- struct [editorConfig](#) [E](#)

### 5.1.1 Detailed Description

The file responsible for running the text editor.

#### Author

Andre Arante

#### Date

6 Jun 2024

### 5.1.2 Macro Definition Documentation

#### 5.1.2.1 \_BSD\_SOURCE

```
#define _BSD_SOURCE
```

#### 5.1.2.2 \_DEFAULT\_SOURCE

```
#define _DEFAULT_SOURCE
```

#### 5.1.2.3 \_GNU\_SOURCE

```
#define _GNU_SOURCE
```

#### 5.1.2.4 ABUF\_INIT

```
#define ABUF_INIT {NULL, 0}
```

#### 5.1.2.5 CTRL\_KEY

```
#define CTRL_KEY(  
    k) ((k) & 0x1f)
```

#### 5.1.2.6 KILO\_QUIT\_TIMES

```
#define KILO_QUIT_TIMES 3
```

#### 5.1.2.7 KILO\_TAB\_STOP

```
#define KILO_TAB_STOP 8
```

### 5.1.2.8 KILO\_VERSION

```
#define KILO_VERSION "0.0.1"
```

## 5.1.3 Typedef Documentation

### 5.1.3.1 erow

```
typedef struct erow erow
```

## 5.1.4 Enumeration Type Documentation

### 5.1.4.1 editorKey

```
enum editorKey
```

Enumerator

BACKSPACE	
ARROW_LEFT	
ARROW_RIGHT	
ARROW_UP	
ARROW_DOWN	
DEL_KEY	
HOME_KEY	
END_KEY	
PAGE_UP	
PAGE_DOWN	

## 5.1.5 Function Documentation

### 5.1.5.1 abAppend()

```
void abAppend (
    struct abuf * ab,
    const char * s,
    int len)
```

### 5.1.5.2 abFree()

```
void abFree (
    struct abuf * ab)
```

### 5.1.5.3 die()

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

A panic function that is called when an unforeseen error occurs.

A more detailed explanation.

## Parameters

*s	the error that was caught and to be displayed to user
----	---

**5.1.5.4 disableRawMode()**

```
void disableRawMode (  
    void )
```

**5.1.5.5 editorDelChar()**

```
void editorDelChar (  
    void )
```

**5.1.5.6 editorDelRow()**

```
void editorDelRow (  
    int at)
```

**5.1.5.7 editorDrawMessageBar()**

```
void editorDrawMessageBar (  
    struct abuf * ab)
```

**5.1.5.8 editorDrawRows()**

```
void editorDrawRows (  
    struct abuf * ab)
```

**5.1.5.9 editorDrawStatusBar()**

```
void editorDrawStatusBar (  
    struct abuf * ab)
```

**5.1.5.10 editorFreeRow()**

```
void editorFreeRow (  
    erow * row)
```

**5.1.5.11 editorInsertChar()**

```
void editorInsertChar (  
    int c)
```

**5.1.5.12 editorInsertNewline()**

```
void editorInsertNewline (  
    void )
```

**5.1.5.13 editorInsertRow()**

```
void editorInsertRow (  
    int at,  
    char * s,  
    size_t len)
```

**5.1.5.14 editorMoveCursor()**

```
void editorMoveCursor (  
    int key)
```

**5.1.5.15 editorOpen()**

```
void editorOpen (  
    char * filename)
```

**5.1.5.16 editorProcessKeypress()**

```
void editorProcessKeypress (  
    void )
```

**5.1.5.17 editorPrompt()**

```
char * editorPrompt (  
    char * prompt)
```

**5.1.5.18 editorReadKey()**

```
int editorReadKey (  
    void )
```

**5.1.5.19 editorRefreshScreen()**

```
void editorRefreshScreen (  
    void )
```

#### 5.1.5.20 editorRowAppendString()

```
void editorRowAppendString (
    erow * row,
    char * s,
    size_t len)
```

#### 5.1.5.21 editorRowCxToRx()

```
int editorRowCxToRx (
    erow * row,
    int cx)
```

#### 5.1.5.22 editorRowDelChar()

```
void editorRowDelChar (
    erow * row,
    int at)
```

#### 5.1.5.23 editorRowInsertChar()

```
void editorRowInsertChar (
    erow * row,
    int at,
    int c)
```

#### 5.1.5.24 editorRowsToString()

```
char * editorRowsToString (
    int * buflen)
```

#### 5.1.5.25 editorSave()

```
void editorSave (
    void )
```

#### 5.1.5.26 editorScroll()

```
void editorScroll (
    void )
```

#### 5.1.5.27 editorSetStatusMessage()

```
void editorSetStatusMessage (
    const char * fmt,
    ...)
```

#### 5.1.5.28 editorUpdateRow()

```
void editorUpdateRow (
    erow * row)
```

#### 5.1.5.29 enableRawMode()

```
void enableRawMode (
    void )
```

#### 5.1.5.30 getCursorPosition()

```
int getCursorPosition (
    int * rows,
    int * cols)
```

#### 5.1.5.31 getWindowSize()

```
int getWindowSize (
    int * rows,
    int * cols)
```

#### 5.1.5.32 initEditor()

```
void initEditor (
    void )
```

#### 5.1.5.33 main()

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

### 5.1.6 Variable Documentation

#### 5.1.6.1 E

```
struct editorConfig E
```

## 5.2 README.md File Reference