

SDL Console Reference Manual

Generated by Doxygen 1.2.18

Fri Mar 14 12:50:07 2003

Contents

1	SDL Console Main Page	1
2	SDL Console Compound Index	3
2.1	SDL Console Compound List	3
3	SDL Console File Index	5
3.1	SDL Console File List	5
4	SDL Console Class Documentation	7
4.1	BitFont_td Struct Reference	7
4.2	console_information_td Struct Reference	8
5	SDL Console File Documentation	13
5.1	CON_console.h File Reference	13
5.2	DT_drawtext.h File Reference	20

Chapter 1

SDL Console Main Page

1.0.1 Introduction

SDL_Console is a console that can be added to any SDL application. It is similar to Quake and other games consoles. A console is meant to be a very simple way of interacting with a program and executing commands. You can also have more than one console at a time.

1.0.2 Documentation

For a detailed description of all functions see **CON_console.h** (p.13). Remark that functions that have the mark "Internal" are only used internally. There's not much use of calling these functions.

Have Fun!

Author:

Garett Banuk <mongoose@mongeese.org> (Original Version) , Clemens Wacha <reflex-2000@gmx.net> (Version 2.x, Documentation) , Boris Lesner <talanthyr@tuxfamily.org> (Package Maintainer)

Chapter 2

SDL Console Compound Index

2.1 SDL Console Compound List

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

BitFont_td	7
console_information_td	8

Chapter 3

SDL Console File Index

3.1 SDL Console File List

Here is a list of all files with brief descriptions:

CON_console.h	13
DT_drawtext.h	20

Chapter 4

SDL Console Class Documentation

4.1 BitFont_td Struct Reference

```
#include <DT_drawtext.h>
```

Public Attributes

- `SDL_Surface * FontSurface`
- `int CharWidth`
- `int CharHeight`
- `int FontNumber`
- `BitFont_td * NextFont`

4.1.1 Member Data Documentation

4.1.1.1 `SDL_Surface* BitFont_td::FontSurface`

4.1.1.2 `int BitFont_td::CharWidth`

4.1.1.3 `int BitFont_td::CharHeight`

4.1.1.4 `int BitFont_td::FontNumber`

4.1.1.5 `struct BitFont_td* BitFont_td::NextFont`

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

- `DT_drawtext.h`
-

4.2 console_information_td Struct Reference

```
#include <CON_console.h>
```

4.2.1 Detailed Description

This is a struct for each consoles data

Public Attributes

- int **visible**
- int **HideKey**
- char ** **ConsoleLines**
- char ** **CommandLines**
- int **TotalConsoleLines**
- int **ConsoleScrollBack**
- int **TotalCommands**
- int **FontNumber**
- int **LineBuffer**
- int **VChars**
- int **BackX**
- int **BackY**
- char * **Prompt**
- char **Command** [CON_CHARS_PER_LINE]
- char **RCommand** [CON_CHARS_PER_LINE]
- char **LCommand** [CON_CHARS_PER_LINE]
- char **VCommand** [CON_CHARS_PER_LINE]
- int **CursorPos**
- int **Offset**
- int **InsMode**
- SDL_Surface * **ConsoleSurface**
- SDL_Surface * **OutputScreen**
- SDL_Surface * **BackgroundImage**
- SDL_Surface * **InputBackground**
- int **DispX**
- int **DispY**
- unsigned char **ConsoleAlpha**
- int **CommandScrollBack**
- void(* **CmdFunction**)(struct console_information_td *console, char *command)
- char *(* **TabFunction**)(char *command)
- int **FontHeight**
- int **FontWidth**

4.2.2 Member Data Documentation

4.2.2.1 int console_information_td::visible

4.2.2.2 int console_information_td::HideKey

boolean: true if console is visible (use Show/Hide functions to toggle)

4.2.2.3 char console_information_td::ConsoleLines**

the key that can hide the console

4.2.2.4 char console_information_td::CommandLines**

List of all the past lines.

4.2.2.5 int console_information_td::TotalConsoleLines

List of all the past commands.

4.2.2.6 int console_information_td::ConsoleScrollBack

Total number of lines in the console.

4.2.2.7 int console_information_td::TotalCommands

How much the users scrolled back in the console.

4.2.2.8 int console_information_td::FontNumber

Number of commands in the Back Commands.

4.2.2.9 int console_information_td::LineBuffer

This is the number of the font for the console.

4.2.2.10 int console_information_td::VChars

The number of visible lines in the console (autocalculated).

4.2.2.11 int console_information_td::BackX

The number of visible characters in one console line (autocalculated).

4.2.2.12 int console_information_td::BackY

The number of visible characters in one console line (autocalculated).

4.2.2.13 char* console_information_td::Prompt

Background images x and y coords.

4.2.2.14 char console_information_td::Command[CON_CHARS_PER_LINE]

Prompt displayed in command line.

4.2.2.15 char console_information_td::RCommand[CON_CHARS_PER_LINE]

current command in command line = lcommand + rcommand

4.2.2.16 char console_information_td::LCommand[CON_CHARS_PER_LINE]

left hand side of cursor

4.2.2.17 char console_information_td::VCommand[CON_CHARS_PER_LINE]

right hand side of cursor

4.2.2.18 int console_information_td::CursorPos

current visible command line

4.2.2.19 int console_information_td::Offset

Current cursor position in CurrentCommand.

4.2.2.20 int console_information_td::InsMode

CommandOffset (first visible char of command) - if command is too long to fit into console.

4.2.2.21 SDL_Surface* console_information_td::ConsoleSurface

Insert or Overwrite characters?

4.2.2.22 SDL_Surface* console_information_td::OutputScreen

Surface of the console.

4.2.2.23 SDL_Surface* console_information_td::BackgroundImage

This is the screen to draw the console to.

4.2.2.24 SDL_Surface* console_information_td::InputBackground

Background image for the console.

4.2.2.25 int console_information_td::DispX

Dirty rectangle to draw over behind the users background.

4.2.2.26 int console_information_td::DispY

Dirty rectangle to draw over behind the users background.

4.2.2.27 unsigned char console_information_td::ConsoleAlpha

The top left x and y coords of the console on the display screen.

4.2.2.28 int console_information_td::CommandScrollBack

The consoles alpha level.

4.2.2.29 void(* console_information_td::CmdFunction)(struct console_information_td *console, char* command)

How much the users scrolled back in the command lines.

4.2.2.30 char*(* console_information_td::TabFunction)(char* command)

The Function that is executed if you press <Return> in the console.

4.2.2.31 int console_information_td::FontHeight

The Function that is executed if you press <Tab> in the console.

4.2.2.32 int console_information_td::FontWidth

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

- CON_console.h

Chapter 5

SDL Console File Documentation

5.1 CON_console.h File Reference

```
#include "SDL.h"
```

Compounds

- struct `console_information_t`

Defines

- `#define CON_CHARS_PER_LINE 128`
- `#define CON_BLINK_RATE 500`
- `#define CON_CHAR_BORDER 4`
- `#define CON_DEFAULT_PROMPT "]"`
- `#define CON_LINE_SCROLL 2`
- `#define CON_SCROLL_INDICATOR " ^"`
- `#define CON_INS_CURSOR " _"`
- `#define CON_OVR_CURSOR " |"`
- `#define CON_DEFAULT_HIDEKEY SDLK_ESCAPE`

Typedefs

- `typedef console_information_t ConsoleInformation`

Functions

- `SDL_Event * CON_Events (SDL_Event *event)`
 - `void CON_Show (ConsoleInformation *console)`
 - `void CON_Hide (ConsoleInformation *console)`
 - `int CON_IsVisible (ConsoleInformation *console)`
 - `void CON_DrawConsole (ConsoleInformation *console)`
 - `ConsoleInformation * CON_Init (const char *FontName, SDL_Surface *DisplayScreen, int lines, SDL_Rect rect)`
-

- void **CON_Destroy** (**ConsoleInformation** *console)
- void **CON_Free** (**ConsoleInformation** *console)
- void **CON_Out** (**ConsoleInformation** *console, const char *str,...)
- void **CON_Alpha** (**ConsoleInformation** *console, unsigned char alpha)
- void **CON_AlphaGL** (SDL_Surface *s, int alpha)
- int **CON_Background** (**ConsoleInformation** *console, const char *image, int x, int y)
- void **CON_Position** (**ConsoleInformation** *console, int x, int y)
- int **CON_Resize** (**ConsoleInformation** *console, SDL_Rect rect)
- void **CON_Topmost** (**ConsoleInformation** *console)
- void **CON_SetPrompt** (**ConsoleInformation** *console, char *newprompt)
- void **CON_SetHideKey** (**ConsoleInformation** *console, int key)
- void **CON_Execute** (**ConsoleInformation** *console, char *command)
- void **CON_SetExecuteFunction** (**ConsoleInformation** *console, void(*Cmd-Function)(**ConsoleInformation** *console2, char *command))
- void **CON_SetTabCompletion** (**ConsoleInformation** *console, char *(*Tab-Function)(char *command))
- void **CON_TabCompletion** (**ConsoleInformation** *console)
- void **CON_NewLineConsole** (**ConsoleInformation** *console)
- void **CON_NewLineCommand** (**ConsoleInformation** *console)
- void **CON_UpdateConsole** (**ConsoleInformation** *console)
- void **Default_CmdFunction** (**ConsoleInformation** *console, char *command)
- char * **Default_TabFunction** (char *command)
- void **DrawCommandLine** ()
- void **Cursor_Left** (**ConsoleInformation** *console)
- void **Cursor_Right** (**ConsoleInformation** *console)
- void **Cursor_Home** (**ConsoleInformation** *console)
- void **Cursor_End** (**ConsoleInformation** *console)
- void **Cursor_Del** (**ConsoleInformation** *console)
- void **Cursor_BSpace** (**ConsoleInformation** *console)
- void **Cursor_Add** (**ConsoleInformation** *console, SDL_Event *event)
- void **Clear_Command** (**ConsoleInformation** *console)
- void **Clear_History** (**ConsoleInformation** *console)
- void **Command_Up** (**ConsoleInformation** *console)
- void **Command_Down** (**ConsoleInformation** *console)

5.1.1 Define Documentation

5.1.1.1 #define CON_CHARS_PER_LINE 128

Cut the buffer line if it becomes longer than this.

5.1.1.2 #define CON_BLINK_RATE 500

Cursor blink frequency in ms.

5.1.1.3 #define CON_CHAR_BORDER 4

Border in pixels from the most left to the first letter.

5.1.1.4 `#define CON_DEFAULT_PROMPT "]"`

Default prompt used at the commandline.

5.1.1.5 `#define CON_LINE_SCROLL 2`

Scroll this many lines at a time (when pressing PGUP or PGDOWN).

5.1.1.6 `#define CON_SCROLL_INDICATOR "^"`

Indicator showing that you scrolled up the history.

5.1.1.7 `#define CON_INS_CURSOR "_"`

Cursor shown if we are in insert mode.

5.1.1.8 `#define CON_OVR_CURSOR "|"`

Cursor shown if we are in overwrite mode.

5.1.1.9 `#define CON_DEFAULT_HIDEKEY SDLK_ESCAPE`

Defines the default hide key (Hide() the console if pressed).

5.1.2 Typedef Documentation

5.1.2.1 `typedef struct console_information_td ConsoleInformation`

This is a struct for each consoles data

5.1.3 Function Documentation

5.1.3.1 `SDL_Event* CON_Events (SDL_Event * event)`

Takes keys from the keyboard and inputs them to the console if the console isVisible(). If the event was not handled (i.e. WM events or unknown ctrl-shift sequences) the function returns the event for further processing.

5.1.3.2 `void CON_Show (ConsoleInformation * console)`

Makes the console visible

5.1.3.3 `void CON_Hide (ConsoleInformation * console)`

Hides the console

5.1.3.4 int CON_IsVisible (ConsoleInformation * *console*)

Return 1 if the console is visible, 0 else

5.1.3.5 void CON_DrawConsole (ConsoleInformation * *console*)

Draws the console to the screen if it isVisible()

5.1.3.6 ConsoleInformation* CON_Init (const char * *FontName*, SDL_Surface * *DisplayScreen*, int *lines*, SDL_Rect *rect*)

Initializes a new console

5.1.3.7 void CON_Destroy (ConsoleInformation * *console*)

Frees DT_DrawText and calls CON_Free

5.1.3.8 void CON_Free (ConsoleInformation * *console*)

Frees all the memory loaded by the console

5.1.3.9 void CON_Out (ConsoleInformation * *console*, const char * *str*, ...)

printf for the console

5.1.3.10 void CON_Alpha (ConsoleInformation * *console*, unsigned char *alpha*)

Sets the alpha channel of an SDL_Surface to the specified value (0 - transparent, 255 - opaque)

5.1.3.11 void CON_AlphaGL (SDL_Suface * *s*, int *alpha*)

Sets the alpha channel of an SDL_Surface to the specified value. Use this for OpenGL stuff. Preconditions: the surface in question is RGBA. $0 \leq a \leq 255$, where 0 is transparent and 255 opaque

5.1.3.12 int CON_Background (ConsoleInformation * *console*, const char * *image*, int *x*, int *y*)

Sets a background image for the consoel

5.1.3.13 void CON_Position (ConsoleInformation * *console*, int *x*, int *y*)

Changes current position of the console

5.1.3.14 int CON_Resize (ConsoleInformation * *console*, SDL_Rect *rect*)

Changes the size of the console

5.1.3.15 void CON_Topmost (ConsoleInformation * *console*)

Give focus to a console. Make it the "topmost" console. This console will receive events sent with **CON_Events()** (p.15)

5.1.3.16 void CON_SetPrompt (ConsoleInformation * *console*, char * *newprompt*)

Modify the prompt of the console

5.1.3.17 void CON_SetHideKey (ConsoleInformation * *console*, int *key*)

Set the key, that invokes a **CON_Hide()** (p.15) after press. default is ESCAPE and you can always hide using ESCAPE and the HideKey. compared against event->key.keysym.sym !!

5.1.3.18 void CON_Execute (ConsoleInformation * *console*, char * *command*)

Internal: executes the command typed in at the console (called if you press ENTER)

5.1.3.19 void CON_SetExecuteFunction (ConsoleInformation * *console*, void(* *CmdFunction*)(ConsoleInformation **console2*, char **command*))

Sets the callback function that is called if a command was typed in. The function could look like this: void my_command_handler(ConsoleInformation* console, char* command).

Parameters:

console: the console the command came from.

command: the command string that was typed in.

5.1.3.20 void CON_SetTabCompletion (ConsoleInformation * *console*, char *(* *TabFunction*)(char **command*))

Sets the callback tabulator completion function. char* my_tabcompletion(char* command). If Tab is pressed, the function gets called with the already typed in command. my_tabcompletion then checks if it can complete the command or if it should display a list of all matching commands (with **CON_Out()** (p.16)). Returns the completed command or NULL if no completion was made.

5.1.3.21 void CON_TabCompletion (ConsoleInformation * *console*)

Internal: Gets called when TAB was pressed

5.1.3.22 void CON_NewLineConsole (ConsoleInformation * *console*)

Internal: makes newline (same as printf("

") or CON_Out(console, "

"))

5.1.3.23 void CON_NewLineCommand (ConsoleInformation * *console*)

Internal: shift command history (the one you can switch with the up/down keys)

5.1.3.24 void CON_UpdateConsole (ConsoleInformation * *console*)

Internal: updates console after resize etc.

5.1.3.25 void Default_CmdFunction (ConsoleInformation * *console*, char * *command*)

Internal: Default Execute callback

5.1.3.26 char* Default_TabFunction (char * *command*)

Internal: Default TabCompletion callback

5.1.3.27 void DrawCommandLine ()

Internal: draws the commandline the user is typing in to the screen. called by update?

5.1.3.28 void Cursor_Left (ConsoleInformation * *console*)

Internal: Gets called if you press the LEFT key (move cursor left)

5.1.3.29 void Cursor_Right (ConsoleInformation * *console*)

Internal: Gets called if you press the RIGHT key (move cursor right)

5.1.3.30 void Cursor_Home (ConsoleInformation * *console*)

Internal: Gets called if you press the HOME key (move cursor to the beginning of the line)

5.1.3.31 void Cursor_End (ConsoleInformation * *console*)

Internal: Gets called if you press the END key (move cursor to the end of the line)

5.1.3.32 void Cursor_Del (ConsoleInformation * *console*)

Internal: Called if you press DELETE (deletes character under the cursor)

5.1.3.33 void Cursor_BSpace (ConsoleInformation * *console*)

Internal: Called if you press BACKSPACE (deletes character left of cursor)

5.1.3.34 void Cursor_Add (ConsoleInformation * *console*, SDL_Event * *event*)

Internal: Called if you type in a character (add the char to the command)

5.1.3.35 void Clear_Command (ConsoleInformation * *console*)

Internal: Called if you press Ctrl-C (deletes the commandline)

5.1.3.36 void Clear_History (ConsoleInformation * *console*)

Internal: Called if you press Ctrl-L (deletes the History)

5.1.3.37 void Command_Up (ConsoleInformation * *console*)

Internal: Called if you press UP key (switches through recent typed in commands)

5.1.3.38 void Command_Down (ConsoleInformation * *console*)

Internal: Called if you press DOWN key (switches through recent typed in commands)

5.2 DT_drawtext.h File Reference

Compounds

- struct **BitFont_td**

Defines

- `#define TRANS_FONT 1`

Typedefs

- `typedef BitFont_td BitFont`

Functions

- `void DT_DrawText (const char *string, SDL_Surface *surface, int FontType, int x, int y)`
- `int DT_LoadFont (const char *BitmapName, int flags)`
- `int DT_FontHeight (int FontNumber)`
- `int DT_FontWidth (int FontNumber)`
- `BitFont * DT_FontPointer (int FontNumber)`
- `void DT_DestroyDrawText ()`

5.2.1 Define Documentation

5.2.1.1 `#define TRANS_FONT 1`

5.2.2 Typedef Documentation

5.2.2.1 `typedef struct BitFont_td BitFont`

5.2.3 Function Documentation

5.2.3.1 `void DT_DrawText (const char * string, SDL_Surface * surface, int FontType, int x, int y)`

5.2.3.2 `int DT_LoadFont (const char * BitmapName, int flags)`

5.2.3.3 `int DT_FontHeight (int FontNumber)`

5.2.3.4 `int DT_FontWidth (int FontNumber)`

5.2.3.5 `BitFont* DT_FontPointer (int FontNumber)`

5.2.3.6 `void DT_DestroyDrawText ()`

Index

BackgroundImage
 console_information_td, 10
BackX
 console_information_td, 9
BackY
 console_information_td, 9
BitFont
 DT_drawtext.h, 20
BitFont_td, 7
BitFont_td
 CharHeight, 7
 CharWidth, 7
 FontNumber, 7
 FontSurface, 7
 NextFont, 7

CharHeight
 BitFont_td, 7
CharWidth
 BitFont_td, 7
Clear_Command
 CON_console.h, 19
Clear_History
 CON_console.h, 19
CmdFunction
 console_information_td, 11
Command
 console_information_td, 9
Command_Down
 CON_console.h, 19
Command_Up
 CON_console.h, 19
CommandLines
 console_information_td, 9
CommandScrollBack
 console_information_td, 11
CON_Alpha
 CON_console.h, 16
CON_AlphaGL
 CON_console.h, 16
CON_Background
 CON_console.h, 16
CON_BLINK_RATE
 CON_console.h, 14
CON_CHAR_BORDER
 CON_console.h, 14
CON_CHARS_PER_LINE
 CON_console.h, 14
CON_console.h, 13
 Clear_Command, 19
 Clear_History, 19
 Command_Down, 19
 Command_Up, 19
 CON_Alpha, 16
 CON_AlphaGL, 16
 CON_Background, 16
 CON_BLINK_RATE, 14
 CON_CHAR_BORDER, 14
 CON_CHARS_PER_LINE, 14
 CON_DEFAULT_HIDEKEY, 15
 CON_DEFAULT_PROMPT, 14
 CON_Destroy, 16
 CON_DrawConsole, 16
 CON_Events, 15
 CON_Execute, 17
 CON_Free, 16
 CON_Hide, 15
 CON_Init, 16
 CON_INS_CURSOR, 15
 CON_isVisible, 15
 CON_LINE_SCROLL, 15
 CON_NewLineCommand, 17
 CON_NewLineConsole, 17
 CON_Out, 16
 CON_OVR_CURSOR, 15
 CON_Position, 16
 CON_Resize, 16
 CON_SCROLL_INDICATOR, 15
 CON_SetExecuteFunction, 17
 CON_SetHideKey, 17
 CON_SetPrompt, 17
 CON_SetTabCompletion, 17
 CON_Show, 15
 CON_TabCompletion, 17
 CON_Topmost, 16
 CON_UpdateConsole, 18
 ConsoleInformation, 15
 Cursor_Add, 18
 Cursor_BSpace, 18

- Cursor_Del, 18
- Cursor_End, 18
- Cursor_Home, 18
- Cursor_Left, 18
- Cursor_Right, 18
- Default_CmdFunction, 18
- Default_TabFunction, 18
- DrawCommandLine, 18
- CON_DEFAULT_HIDEKEY
 - CON_console.h, 15
- CON_DEFAULT_PROMPT
 - CON_console.h, 14
- CON_Destroy
 - CON_console.h, 16
- CON_DrawConsole
 - CON_console.h, 16
- CON_Events
 - CON_console.h, 15
- CON_Execute
 - CON_console.h, 17
- CON_Free
 - CON_console.h, 16
- CON_Hide
 - CON_console.h, 15
- CON_Init
 - CON_console.h, 16
- CON_INS_CURSOR
 - CON_console.h, 15
- CON_isVisible
 - CON_console.h, 15
- CON_LINE_SCROLL
 - CON_console.h, 15
- CON_NewLineCommand
 - CON_console.h, 17
- CON_NewLineConsole
 - CON_console.h, 17
- CON_Out
 - CON_console.h, 16
- CON_OVR_CURSOR
 - CON_console.h, 15
- CON_Position
 - CON_console.h, 16
- CON_Resize
 - CON_console.h, 16
- CON_SCROLL_INDICATOR
 - CON_console.h, 15
- CON_SetExecuteFunction
 - CON_console.h, 17
- CON_SetHideKey
 - CON_console.h, 17
- CON_SetPrompt
 - CON_console.h, 17
- CON_SetTabCompletion
 - CON_console.h, 17
- CON_Show
 - CON_console.h, 15
- CON_TabCompletion
 - CON_console.h, 17
- CON_Topmost
 - CON_console.h, 16
- CON_UpdateConsole
 - CON_console.h, 18
- console_information_td, 8
 - BackgroundImage, 10
 - BackX, 9
 - BackY, 9
 - CmdFunction, 11
 - Command, 9
 - CommandLines, 9
 - CommandScrollBack, 11
 - ConsoleAlpha, 10
 - ConsoleLines, 8
 - ConsoleScrollBack, 9
 - ConsoleSurface, 10
 - CursorPos, 10
 - DispX, 10
 - DispY, 10
 - FontHeight, 11
 - FontNumber, 9
 - FontWidth, 11
 - HideKey, 8
 - InputBackground, 10
 - InsMode, 10
 - LCommand, 10
 - LineBuffer, 9
 - Offset, 10
 - OutputScreen, 10
 - Prompt, 9
 - RCommand, 9
 - TabFunction, 11
 - TotalCommands, 9
 - TotalConsoleLines, 9
 - VChars, 9
 - VCommand, 10
 - visible, 8
- ConsoleAlpha
 - console_information_td, 10
- ConsoleInformation
 - CON_console.h, 15
- ConsoleLines
 - console_information_td, 8
- ConsoleScrollBack
 - console_information_td, 9
- ConsoleSurface
 - console_information_td, 10
- Cursor_Add
 - CON_console.h, 18
- Cursor_BSpace

- CON_console.h, 18
- Cursor_Del
 - CON_console.h, 18
- Cursor_End
 - CON_console.h, 18
- Cursor_Home
 - CON_console.h, 18
- Cursor_Left
 - CON_console.h, 18
- Cursor_Right
 - CON_console.h, 18
- CursorPos
 - console_information_td, 10
- Default_CmdFunction
 - CON_console.h, 18
- Default_TabFunction
 - CON_console.h, 18
- DispX
 - console_information_td, 10
- DispY
 - console_information_td, 10
- DrawCommandLine
 - CON_console.h, 18
- DT_DestroyDrawText
 - DT_drawtext.h, 20
- DT_DrawText
 - DT_drawtext.h, 20
- DT_drawtext.h, 20
 - BitFont, 20
 - DT_DestroyDrawText, 20
 - DT_DrawText, 20
 - DT_FontHeight, 20
 - DT_FontPointer, 20
 - DT_FontWidth, 20
 - DT_LoadFont, 20
 - TRANS_FONT, 20
- DT_FontHeight
 - DT_drawtext.h, 20
- DT_FontPointer
 - DT_drawtext.h, 20
- DT_FontWidth
 - DT_drawtext.h, 20
- DT_LoadFont
 - DT_drawtext.h, 20
- FontHeight
 - console_information_td, 11
- FontNumber
 - BitFont_td, 7
 - console_information_td, 9
- FontSurface
 - BitFont_td, 7
- FontWidth
 - console_information_td, 11
- HideKey
 - console_information_td, 8
- InputBackground
 - console_information_td, 10
- InsMode
 - console_information_td, 10
- LCommand
 - console_information_td, 10
- LineBuffer
 - console_information_td, 9
- NextFont
 - BitFont_td, 7
- Offset
 - console_information_td, 10
- OutputScreen
 - console_information_td, 10
- Prompt
 - console_information_td, 9
- RCommand
 - console_information_td, 9
- TabFunction
 - console_information_td, 11
- TotalCommands
 - console_information_td, 9
- TotalConsoleLines
 - console_information_td, 9
- TRANS_FONT
 - DT_drawtext.h, 20
- VChars
 - console_information_td, 9
- VCommand
 - console_information_td, 10
- visible
 - console_information_td, 8