

libdaisy Reference Manual

0.2.2

Generated by Doxygen 1.4.2

Thu May 18 12:46:33 2006

Contents

1 Libdaisy	1
2 libdaisy Data Structure Index	2
3 libdaisy File Index	2
4 libdaisy Data Structure Documentation	3
5 libdaisy File Documentation	3

1 Libdaisy

1.1 Introduction

Libdaisy is a toolkit for parsing and playing Daisy Digital Talking Books (DTB). Libdaisy is developed for the Linux operation systems under the [GNU General Public License](#).

Libdaisy does not offer the complete functionality according to the Daisy standards and is not a finished library. Even though the library is not complete, it offers the most important functionality for playing a Daisy DTB. Libdaisy partially supports two of the Daisy standards, [DAISY 2.02](#) and the new standard [ANSI/NISO Z39.86](#).

Please visit the [DaisyPlayer](#) project web page for more information.

1.2 Requirements

- libxml2 >=2.6.16-7
- libao2 >=0.8.6-1
- libmad0 >=0.15.1b-1.1

1.3 Install

From package:

Install the Debian package, using `'dpkg -i libdaisy_0.2.0_i386.deb'`.

From source:

```
'tar -zxvf libdaisy_0.2.0.tar.gz'
```

```
'cd libdaisy_0.2.0'
```

```
'make'
```

```
'make install' or 'make install DESTDIR=/tmp/foo' #'make install' needs root privilege
```

```
'make install-dev' #'install the development header, needs root privilege
```

1.4 License

Copyright (C) 2006 by Andr 169 Lindhjem <belgarat@sdf.lonestar.org>, Kjetil Holien <kjetil.holien@gmail.com>, Terje Risa <terje.risa@gmail.com> &  152yvind Nerbr 165ten <oyvind@nerbraten.com>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

1.5 Bug Reports

Any kind of bug reports are welcome. If you find bugs, please email us (See the Author section for contact information).

Author:

Andr 169 Lindhjem <belgarat@sdf.lonestar.org>
Kjetil Holien <kjetil.holien@gmail.com>
Terje Risa <terje.risa@gmail.com>
 152yvind Nerbr 165ten <oyvind@nerbraten.com>

Date:

26.04.2006

Version:

0.2.2

2 libdaisy Data Structure Index

2.1 libdaisy Data Structures

Here are the data structures with brief descriptions:

[daisy_position](#) (Struct for holding a Daisy DTB playback position) 3

3 libdaisy File Index

3.1 libdaisy File List

Here is a list of all files with brief descriptions:

[libdaisy.h](#) (Header file for the libdaisy development package) 3

[libdaisy_mainpage.doxygen](#) 10

4 libdaisy Data Structure Documentation

4.1 daisy_position Struct Reference

4.1.1 Detailed Description

Struct for holding a Daisy DTB playback position.

Meant for storing chapter (smilpos) and passage (nodepos) jump positions. Works as a bookmark. The integer values is equal to the chapter number and passage number in the playback sequence (smilpos = 1 & nodepos = 1 is the position for passage 1 in chapter 1).

See also:

[daisy_get_position](#)
[daisy_goto_position](#)

Data Fields

- int [smilpos](#)
- int [nodepos](#)

4.1.2 Field Documentation

4.1.2.1 int [daisy_position::nodepos](#)

4.1.2.2 int [daisy_position::smilpos](#)

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

- [libdaisy.h](#)

5 libdaisy File Documentation

5.1 libdaisy.h File Reference

5.1.1 Detailed Description

Header file for the libdaisy development package.

Libdaisy is a toolkit for parsing and playing Daisy Digital Talking Books (DTB).

Date:

21.03.2006

Typedefs

- typedef void * [daisyplayer_t](#)

This will need to be initialized and passed along to the daisy functions.

Enumerations

- enum `daisy_status` {
 - `DAISY_ERROR_UNKNOWN` = 0x0000, `DAISY_ERROR_AUDIO_NOT_INITIALIZED` = 0x0200, `DAISY_ERROR_AUDIO_CREATE_MMAP` = 0x0201, `DAISY_ERROR_AUDIO_FSTAT` = 0x0202,
 - `DAISY_ERROR_AUDIO_OPEN` = 0x0203, `DAISY_ERROR_AUDIO_FREE_MMAP` = 0x0204, `DAISY_ERROR_AUDIO_INITIATE_DATA` = 0x0205, `DAISY_ERROR_AUDIO_NOT_PLAYING` = 0x0206,
 - `DAISY_ERROR_AUDIO_NOT_STOPPED` = 0x0207, `DAISY_ERROR_AUDIO_DATA_IS_NULL` = 0x0208, `DAISY_ERROR_AUDIO_MALLOC` = 0x0209, `DAISY_ERROR_AUDIO_PAUSED_WHILE_NOT_PLAYING` = 0x0210,
 - `DAISY_ERROR_AUDIO_STOPPED_WHILE_NOT_PLAYING` = 0x0211, `DAISY_ERROR_PLAYBACK_NO_TEXT_IN_SEGMENT` = 0x0301, `DAISY_ERROR_PLAYBACK_NO_AUDIO_IN_SEGMENT` = 0x0302, `DAISY_ERROR_PLAYBACK_NO_DTB_LOADED` = 0x0303,
 - `DAISY_ERROR_PLAYBACK_SEEK_FAILED` = 0x0304, `DAISY_ERROR_MISC_INIT_MUTEX` = 0x0400, `DAISY_END_OF_BOOK` = 0x1100 }

The different sort of status messages libdaisy might return.

- enum `daisy_seek_option` {
 - `DAISY_SEEK_PREV_CHAPTER` = 1, `DAISY_SEEK_PREV_PASSAGE` = 2, `DAISY_SEEK_NEXT_CHAPTER` = 3, `DAISY_SEEK_NEXT_PASSAGE` = 4,
 - `DAISY_SEEK_TO_BEGINNING` = 5 }

The different seek operations supported by the engine.

- enum `daisy_bookinfo_option` { `DAISY_BOOKINFO_TITLETEXT` = 1, `DAISY_BOOKINFO_TITLEIMAGE` = 2, `DAISY_BOOKINFO_TOTALTIME` = 3 }

The different types of book information.

- enum `daisy_chapter_info` { `DAISY_CHAPTER_TITLE` = 1, `DAISY_CHAPTER_WEIGHT` = 2 }

The different types of chapter information.

Functions

- `daisyplayer_t daisy_init` (void *data, void(*l_cb_daisy_audio_done)(void *), void(*l_cb_daisy_audio_next)(void *, unsigned long int), void(*l_cb_daisy_text)(void *, void *), void(*l_cb_daisy_id)(void *, void *), void(*l_cb_daisy_error)(void *, enum `daisy_status`, const char *daisy_status_msg), void(*l_cb_daisy_progress)(void *, long int))

Initializes the daisy library.

- void `daisy_term` (`daisyplayer_t` daisy)

Terminates the daisy library and frees memory used by it.

- int `daisy_load` (`daisyplayer_t` daisy, char *path)

Loads a new daisy book.

- int `daisy_play` (`daisyplayer_t` daisy)

Starts playback if a book is loaded.

- int [daisy_seek](#) ([daisyplayer_t](#) daisy, int seek_option)
Seek operations which can be performed when a book is loaded.
- [daisy_position](#) * [daisy_get_position](#) ([daisyplayer_t](#) daisy)
Retrieves the current playback position.
- int [daisy_goto_position](#) ([daisyplayer_t](#) daisy, [daisy_position](#) *position)
Seeks to a playback position (bookmark) and continue playback from there.
- int [daisy_stop](#) ([daisyplayer_t](#) daisy)
Stops playback.
- int [daisy_pause](#) ([daisyplayer_t](#) daisy)
Toggle pause.
- char * [daisy_get_info](#) ([daisyplayer_t](#) daisy, int value)
Retrieves book meta information.
- int [daisy_get_chapter_count](#) ([daisyplayer_t](#) daisy)
Retrieves the number of chapters in in the loaded Daisy DTB.
- char * [daisy_get_chapter_info](#) ([daisyplayer_t](#) daisy, int num, int option)
Retrieves information about a given chapter.

5.1.2 Typedef Documentation

5.1.2.1 typedef void* [daisyplayer_t](#)

This will need to be initialized and passed along to the daisy functions.

5.1.3 Enumeration Type Documentation

5.1.3.1 enum [daisy_bookinfo_option](#)

The different types of book information.

See also:

[daisy_get_info](#)

Enumeration values:

DAISY_BOOKINFO_TITLETEXT

DAISY_BOOKINFO_TITLEIMAGE

DAISY_BOOKINFO_TOTALTIME

5.1.3.2 enum [daisy_chapter_info](#)

The different types of chapter information.

See also:

[daisy_get_chapter_info](#)

Enumeration values:

DAISY_CHAPTER_TITLE
DAISY_CHAPTER_WEIGHT

5.1.3.3 enum [daisy_seek_option](#)

The different seek operations supported by the engine.

See also:

[daisy_seek](#)

Enumeration values:

DAISY_SEEK_PREV_CHAPTER
DAISY_SEEK_PREV_PASSAGE
DAISY_SEEK_NEXT_CHAPTER
DAISY_SEEK_NEXT_PASSAGE
DAISY_SEEK_TO_BEGINNING

5.1.3.4 enum [daisy_status](#)

The different sort of status messages libdaisy might return.

Enumeration values:

DAISY_ERROR_UNKNOWN
DAISY_ERROR_AUDIO_NOT_INITIALIZED
DAISY_ERROR_AUDIO_CREATE_MMAP
DAISY_ERROR_AUDIO_FSTAT
DAISY_ERROR_AUDIO_OPEN
DAISY_ERROR_AUDIO_FREE_MMAP
DAISY_ERROR_AUDIO_INITIATE_DATA
DAISY_ERROR_AUDIO_NOT_PLAYING
DAISY_ERROR_AUDIO_NOT_STOPPED
DAISY_ERROR_AUDIO_DATA_IS_NULL
DAISY_ERROR_AUDIO_MALLOC
DAISY_ERROR_AUDIO_PAUSED_WHILE_NOT_PLAYING
DAISY_ERROR_AUDIO_STOPPED_WHILE_NOT_PLAYING
DAISY_ERROR_PLAYBACK_NO_TEXT_IN_SEGMENT
DAISY_ERROR_PLAYBACK_NO_AUDIO_IN_SEGMENT
DAISY_ERROR_PLAYBACK_NO_DTB_LOADED
DAISY_ERROR_PLAYBACK_SEEK_FAILED
DAISY_ERROR_MISC_INIT_MUTEX
DAISY_END_OF_BOOK

5.1.4 Function Documentation

5.1.4.1 `int daisy_get_chapter_count (daisyplayer_t daisy)`

Retrieves the number of chapters in the loaded Daisy DTB.

Parameters:

daisy - the daisy struct which must be passed along with all the API functions.

Returns:

the number of chapters, or -1 in case of error.

5.1.4.2 `char* daisy_get_chapter_info (daisyplayer_t daisy, int num, int option)`

Retrieves information about a given chapter.

Parameters:

daisy - the daisy struct which must be passed along with all the API functions.

num - the chapter number to retrieve information from (use `daisy_get_chapter_count` to get the number of chapters available).

option - a `daisy_chapter_info` which states what information to retrieve.

Returns:

a pointer to a string containing the information, or NULL in case of error.

See also:

[daisy_chapter_info](#)

[daisy_get_chapter_count](#)

5.1.4.3 `char* daisy_get_info (daisyplayer_t daisy, int value)`

Retrieves book meta information.

Parameters:

daisy - the daisy struct which must be passed along with all the API functions.

value - a `daisy_bookinfo_option` value which states what information to retrieve.

Returns:

the string if found, otherwise NULL. The string must be deallocated by the caller.

See also:

[daisy_bookinfo_option](#)

5.1.4.4 `daisy_position* daisy_get_position (daisyplayer_t daisy)`

Retrieves the current playback position.

Parameters:

daisy - the daisy data struct which must be passed along with all the API functions.

Returns:

a pointer to a [daisy_position](#) struct containing the chapter and passage positions, or NULL in case of error. The struct must be deallocated by the caller.

See also:

[daisy_position](#)
[daisy_goto_position](#)

5.1.4.5 int daisy_goto_position (daisyplayer_t daisy, daisy_position * position)

Seeks to a playback position (bookmark) and continue playback from there.

Parameters:

daisy - the daisy data struct which must be passed along with all the API functions.
position - a pointer to a [daisy_position](#) struct with the chapter and passage position.

Returns:

1 in case of success and -1 in case of error.

See also:

[daisy_position](#)
[daisy_get_position](#)

5.1.4.6 daisyplayer_t daisy_init (void * data, void(*)(void *) l_cb_daisy_audio_done, void(*)(void *, unsigned long int) l_cb_daisy_audio_next, void(*)(void *, void *) l_cb_daisy_text, void(*)(void *, void *) l_cb_daisy_id, void(*)(void *, enum daisy_status, const char *daisy_status_msg) l_cb_daisy_error, void(*)(void *, long int) l_cb_daisy_progress)

Initializes the daisy library.

It should be called before any attempt to use the daisy functionality.

Parameters:

data - a void pointer to any object or datastructure you may need in the callback functions. This data pointer will be available in all callback functions. You can e.g. pass along a GUI object in c++ so that you can output the text from the callbackfunctions in the GUI. Set this parameter to NULL if you don't need it.

l_cb_daisy_audio_done - a pointer to the function which will be called when an audio segment is done playing.

l_cb_daisy_audio_next - a pointer to the function which will be called when an new audio segment starts playing, supplying the duration of the segment in ms.

l_cb_daisy_text - a pointer to the function which will be called when an new audio segment starts playing, supplying the text corresponding to the audio.

l_cb_daisy_id - a pointer to the function which will be called when an new audio segment starts playing, supplying the id of the text passage in the xml file.

l_cb_daisy_error - a pointer to the function which will be called when an engine error occurs.

l_cb_daisy_progress - a pointer to the function which will be called during playback, supplying the progress in ms.

Returns:

[daisyplayer_t](#) - the daisy data struct which must be passed along with all the API functions.

5.1.4.7 int daisy_load ([daisyplayer_t](#) *daisy*, char * *path*)

Loads a new daisy book.

Parameters:

daisy - the daisy data struct which must be passed along with all the API functions.

path - a full path to the daisy dtb (ncc.* | *.ncx) to open.

Returns:

1 in case of success and -1 in case of error.

5.1.4.8 int daisy_pause ([daisyplayer_t](#) *daisy*)

Toggle pause.

Pauses playback if state is playing and continues playing of state is paused.

Parameters:

daisy - the daisy data struct which must be passed along with all the API functions.

Returns:

1 in case of success and -1 in case of error.

5.1.4.9 int daisy_play ([daisyplayer_t](#) *daisy*)

Starts playback if a book is loaded.

Parameters:

daisy - the daisy data struct which must be passed along with all the API functions.

Returns:

1 in case of success and -1 in case of error.

5.1.4.10 int daisy_seek ([daisyplayer_t](#) *daisy*, int *seek_option*)

Seek operations which can be performed when a book is loaded.

Parameters:

daisy - the daisy data struct which must be passed along with all the API functions.

seek_option - a [daisy_seek_option](#).

Returns:

0 if end_of_book, 1 in case of success and -1 in case of error.

See also:

[daisy_seek_option](#)

5.1.4.11 int daisy_stop (daisyplayer_t daisy)

Stops playback.

Parameters:

daisy - the daisy data struct which must be passed along with all the API functions.

Returns:

1 in case of success. Has no other return values at this point.

5.1.4.12 void daisy_term (daisyplayer_t daisy)

Terminates the daisy library and frees memory used by it.

It should be called when daisy are no longer needed.

Parameters:

daisy - the daisy data struct which must be passed along with all the API functions.

5.2 libdaisy_mainpage.doxygen File Reference

Index

daisy_bookinfo_option
libdaisy.h, 5

DAISY_BOOKINFO_TITLEIMAGE
libdaisy.h, 5

DAISY_BOOKINFO_TITLETEXT
libdaisy.h, 5

DAISY_BOOKINFO_TOTALTIME
libdaisy.h, 5

daisy_chapter_info
libdaisy.h, 5

DAISY_CHAPTER_TITLE
libdaisy.h, 5

DAISY_CHAPTER_WEIGHT
libdaisy.h, 5

DAISY_END_OF_BOOK
libdaisy.h, 6

DAISY_ERROR_AUDIO_CREATE_MMAP
libdaisy.h, 6

DAISY_ERROR_AUDIO_DATA_IS_NULL
libdaisy.h, 6

DAISY_ERROR_AUDIO_FREE_MMAP
libdaisy.h, 6

DAISY_ERROR_AUDIO_FSTAT
libdaisy.h, 6

DAISY_ERROR_AUDIO_INITIATE_DATA
libdaisy.h, 6

DAISY_ERROR_AUDIO_MALLOC
libdaisy.h, 6

DAISY_ERROR_AUDIO_NOT_INITIALIZED
libdaisy.h, 6

DAISY_ERROR_AUDIO_NOT_PLAYING
libdaisy.h, 6

DAISY_ERROR_AUDIO_NOT_STOPPED
libdaisy.h, 6

DAISY_ERROR_AUDIO_OPEN
libdaisy.h, 6

DAISY_ERROR_AUDIO_PAUSED_WHILE_-
NOT_PLAYING
libdaisy.h, 6

DAISY_ERROR_AUDIO_STOPPED_-
WHILE_NOT_PLAYING
libdaisy.h, 6

DAISY_ERROR_MISC_INIT_MUTEX
libdaisy.h, 6

DAISY_ERROR_PLAYBACK_NO_AUDIO_-
IN_SEGMENT
libdaisy.h, 6

DAISY_ERROR_PLAYBACK_NO_DTB_-
LOADED
libdaisy.h, 6

DAISY_ERROR_PLAYBACK_NO_TEXT_-
IN_SEGMENT
libdaisy.h, 6

DAISY_ERROR_PLAYBACK_SEEK_-
FAILED
libdaisy.h, 6

DAISY_ERROR_UNKNOWN
libdaisy.h, 6

daisy_get_chapter_count
libdaisy.h, 6

daisy_get_chapter_info
libdaisy.h, 6

daisy_get_info
libdaisy.h, 7

daisy_get_position
libdaisy.h, 7

daisy_goto_position
libdaisy.h, 7

daisy_init
libdaisy.h, 8

daisy_load
libdaisy.h, 8

daisy_pause
libdaisy.h, 8

daisy_play
libdaisy.h, 9

daisy_position, 2
nodepos, 3
smilpos, 3

daisy_seek
libdaisy.h, 9

DAISY_SEEK_NEXT_CHAPTER
libdaisy.h, 6

DAISY_SEEK_NEXT_PASSAGE
libdaisy.h, 6

daisy_seek_option
libdaisy.h, 5

DAISY_SEEK_PREV_CHAPTER
libdaisy.h, 6

DAISY_SEEK_PREV_PASSAGE
libdaisy.h, 6

DAISY_SEEK_TO_BEGINNING
libdaisy.h, 6

daisy_status
libdaisy.h, 6

daisy_stop
libdaisy.h, 9

daisy_term
libdaisy.h, 9

daisyplayer_t

- libdaisy.h, 5
- libdaisy.h, 3
 - daisy_bookinfo_option, 5
 - DAISY_BOOKINFO_TITLEIMAGE, 5
 - DAISY_BOOKINFO_TITLETEXT, 5
 - DAISY_BOOKINFO_TOTALTIME, 5
 - daisy_chapter_info, 5
 - DAISY_CHAPTER_TITLE, 5
 - DAISY_CHAPTER_WEIGHT, 5
 - DAISY_END_OF_BOOK, 6
 - DAISY_ERROR_AUDIO_CREATE_-
MMAP, 6
 - DAISY_ERROR_AUDIO_DATA_IS_-
NULL, 6
 - DAISY_ERROR_AUDIO_FREE_MMAP,
6
 - DAISY_ERROR_AUDIO_FSTAT, 6
 - DAISY_ERROR_AUDIO_INITIATE_-
DATA, 6
 - DAISY_ERROR_AUDIO_MALLOC, 6
 - DAISY_ERROR_AUDIO_NOT_-
INITIALIZED, 6
 - DAISY_ERROR_AUDIO_NOT_-
PLAYING, 6
 - DAISY_ERROR_AUDIO_NOT_-
STOPPED, 6
 - DAISY_ERROR_AUDIO_OPEN, 6
 - DAISY_ERROR_AUDIO_PAUSED_-
WHILE_NOT_PLAYING, 6
 - DAISY_ERROR_AUDIO_STOPPED_-
WHILE_NOT_PLAYING, 6
 - DAISY_ERROR_MISC_INIT_MUTEX, 6
 - DAISY_ERROR_PLAYBACK_NO_-
AUDIO_IN_SEGMENT, 6
 - DAISY_ERROR_PLAYBACK_NO_-
DTB_LOADED, 6
 - DAISY_ERROR_PLAYBACK_NO_-
TEXT_IN_SEGMENT, 6
 - DAISY_ERROR_PLAYBACK_SEEK_-
FAILED, 6
 - DAISY_ERROR_UNKNOWN, 6
 - daisy_get_chapter_count, 6
 - daisy_get_chapter_info, 6
 - daisy_get_info, 7
 - daisy_get_position, 7
 - daisy_goto_position, 7
 - daisy_init, 8
 - daisy_load, 8
 - daisy_pause, 8
 - daisy_play, 9
 - daisy_seek, 9
 - DAISY_SEEK_NEXT_CHAPTER, 6
 - DAISY_SEEK_NEXT_PASSAGE, 6
 - daisy_seek_option, 5
 - DAISY_SEEK_PREV_CHAPTER, 6
 - DAISY_SEEK_PREV_PASSAGE, 6
 - DAISY_SEEK_TO_BEGINNING, 6
 - daisy_status, 6
 - daisy_stop, 9
 - daisy_term, 9
 - daisyplayer_t, 5
- libdaisy_mainpage.doxygen, 10
- nodepos
 - daisy_position, 3
- smilpos
 - daisy_position, 3