

libbluray

Generated by Doxygen 1.9.3



---

<b>1 Deprecated List</b>	<b>1</b>
<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 BD_ARGB_BUFFER Struct Reference . . . . .	7
4.1.1 Detailed Description . . . . .	8
4.1.2 Field Documentation . . . . .	8
4.1.2.1 buf . . . . .	8
4.1.2.2 . . . . .	8
4.2 BD_ARGB_OVERLAY Struct Reference . . . . .	8
4.2.1 Detailed Description . . . . .	9
4.3 BD_DIR_H Struct Reference . . . . .	9
4.3.1 Detailed Description . . . . .	10
4.3.2 Field Documentation . . . . .	10
4.3.2.1 close . . . . .	10
4.3.2.2 read . . . . .	10
4.4 BD_DIRENT Struct Reference . . . . .	10
4.4.1 Detailed Description . . . . .	11
4.5 BD_EVENT Struct Reference . . . . .	11
4.5.1 Detailed Description . . . . .	11
4.6 BD_FILE_H Struct Reference . . . . .	11
4.6.1 Detailed Description . . . . .	12
4.6.2 Field Documentation . . . . .	12
4.6.2.1 close . . . . .	12
4.6.2.2 eof . . . . .	12
4.6.2.3 internal . . . . .	13
4.6.2.4 read . . . . .	13
4.6.2.5 seek . . . . .	13
4.6.2.6 tell . . . . .	14
4.6.2.7 write . . . . .	14
4.7 BD_OVERLAY Struct Reference . . . . .	15
4.7.1 Detailed Description . . . . .	15
4.8 BD_PG_PALETTE_ENTRY Struct Reference . . . . .	16
4.8.1 Detailed Description . . . . .	16
4.8.2 Field Documentation . . . . .	16

---

4.8.2.1 T	16
4.9 BD_PG_RLE_ELEM Struct Reference	17
4.9.1 Detailed Description	17
4.10 BLURAY_CLIP_INFO Struct Reference	17
4.10.1 Detailed Description	18
4.11 BLURAY_DISC_INFO Struct Reference	18
4.11.1 Detailed Description	20
4.11.2 Field Documentation	20
4.11.2.1 bdj_detected	20
4.11.2.2 first_play	21
4.11.2.3 no_menu_support	21
4.11.2.4 top_menu	21
4.12 BLURAY_SOUND_EFFECT Struct Reference	21
4.12.1 Detailed Description	22
4.12.2 Field Documentation	22
4.12.2.1 samples	22
4.13 BLURAY_STREAM_INFO Struct Reference	22
4.13.1 Detailed Description	23
4.14 BLURAY_TITLE Struct Reference	23
4.14.1 Detailed Description	23
4.14.2 Field Documentation	23
4.14.2.1 bdj	24
4.15 BLURAY_TITLE_CHAPTER Struct Reference	24
4.15.1 Detailed Description	24
4.16 BLURAY_TITLE_INFO Struct Reference	24
4.16.1 Detailed Description	25
4.17 BLURAY_TITLE_MARK Struct Reference	25
4.17.1 Detailed Description	26
4.18 META_DL Struct Reference	26
4.18.1 Detailed Description	27
4.19 META_THUMBNAIL Struct Reference	27
4.19.1 Detailed Description	27
4.20 META_TITLE Struct Reference	27
4.20.1 Detailed Description	27
<b>5 File Documentation</b>	<b>29</b>
5.1 filesystem.h File Reference	29
5.1.1 Detailed Description	30
5.1.2 Typedef Documentation	30

---

5.1.2.1 BD_DIR_OPEN . . . . .	30
5.1.2.2 BD_FILE_OPEN . . . . .	30
5.1.3 Function Documentation . . . . .	31
5.1.3.1 bd_register_dir() . . . . .	31
5.1.3.2 bd_register_file() . . . . .	31
5.2 filesystem.h . . . . .	32
5.3 meta_data.h File Reference . . . . .	33
5.3.1 Detailed Description . . . . .	33
5.4 meta_data.h . . . . .	33
5.5 bluray-version.h File Reference . . . . .	34
5.5.1 Detailed Description . . . . .	34
5.5.2 Macro Definition Documentation . . . . .	34
5.5.2.1 BLURAY_VERSION_CODE . . . . .	35
5.6 bluray-version.h . . . . .	35
5.7 bluray.h File Reference . . . . .	35
5.7.1 Detailed Description . . . . .	44
5.7.2 Typedef Documentation . . . . .	44
5.7.2.1 bd_argb_overlay_proc_f . . . . .	44
5.7.2.2 bd_overlay_proc_f . . . . .	44
5.7.2.3 BLURAY . . . . .	44
5.7.3 Enumeration Type Documentation . . . . .	45
5.7.3.1 bd_event_e . . . . .	45
5.7.3.2 bd_mark_type_e . . . . .	46
5.7.3.3 bd_player_setting . . . . .	46
5.7.3.4 bd_still_mode_e . . . . .	47
5.7.3.5 bd_video_format_e . . . . .	47
5.7.3.6 bd_video_rate_e . . . . .	48
5.7.4 Function Documentation . . . . .	48
5.7.4.1 bd_chapter_pos() . . . . .	48
5.7.4.2 bd_close() . . . . .	49
5.7.4.3 bd_event_name() . . . . .	49
5.7.4.4 bd_free_clpi() . . . . .	49
5.7.4.5 bd_free_title_info() . . . . .	50
5.7.4.6 bd_get_clpi() . . . . .	50
5.7.4.7 bd_get_current_angle() . . . . .	50
5.7.4.8 bd_get_current_chapter() . . . . .	51
5.7.4.9 bd_get_current_title() . . . . .	51
5.7.4.10 bd_get_disc_info() . . . . .	52
5.7.4.11 bd_get_event() . . . . .	52

---

5.7.4.12 <a href="#">bd_get_main_title()</a> . . . . .	52
5.7.4.13 <a href="#">bd_get_meta()</a> . . . . .	53
5.7.4.14 <a href="#">bd_get_meta_file()</a> . . . . .	53
5.7.4.15 <a href="#">bd_get_playlist_info()</a> . . . . .	54
5.7.4.16 <a href="#">bd_get_sound_effect()</a> . . . . .	54
5.7.4.17 <a href="#">bd_get_title_info()</a> . . . . .	55
5.7.4.18 <a href="#">bd_get_title_size()</a> . . . . .	55
5.7.4.19 <a href="#">bd_get_titles()</a> . . . . .	56
5.7.4.20 <a href="#">bd_get_version()</a> . . . . .	56
5.7.4.21 <a href="#">bd_init()</a> . . . . .	57
5.7.4.22 <a href="#">bd_menu_call()</a> . . . . .	57
5.7.4.23 <a href="#">bd_mouse_select()</a> . . . . .	57
5.7.4.24 <a href="#">bd_open()</a> . . . . .	58
5.7.4.25 <a href="#">bd_open_dir()</a> . . . . .	58
5.7.4.26 <a href="#">bd_open_disc()</a> . . . . .	59
5.7.4.27 <a href="#">bd_open_file_dec()</a> . . . . .	59
5.7.4.28 <a href="#">bd_open_files()</a> . . . . .	60
5.7.4.29 <a href="#">bd_open_stream()</a> . . . . .	60
5.7.4.30 <a href="#">bd_play()</a> . . . . .	61
5.7.4.31 <a href="#">bd_play_title()</a> . . . . .	61
5.7.4.32 <a href="#">bd_read()</a> . . . . .	62
5.7.4.33 <a href="#">bd_read_ext()</a> . . . . .	62
5.7.4.34 <a href="#">bd_read_file()</a> . . . . .	63
5.7.4.35 <a href="#">bd_read_skip_still()</a> . . . . .	63
5.7.4.36 <a href="#">bd_register_argb_overlay_proc()</a> . . . . .	64
5.7.4.37 <a href="#">bd_register_overlay_proc()</a> . . . . .	64
5.7.4.38 <a href="#">bd_seamless_angle_change()</a> . . . . .	65
5.7.4.39 <a href="#">bd_seek()</a> . . . . .	65
5.7.4.40 <a href="#">bd_seek_chapter()</a> . . . . .	65
5.7.4.41 <a href="#">bd_seek_mark()</a> . . . . .	67
5.7.4.42 <a href="#">bd_seek_playitem()</a> . . . . .	67
5.7.4.43 <a href="#">bd_seek_time()</a> . . . . .	68
5.7.4.44 <a href="#">bd_select_angle()</a> . . . . .	68
5.7.4.45 <a href="#">bd_select_playlist()</a> . . . . .	69
5.7.4.46 <a href="#">bd_select_stream()</a> . . . . .	69
5.7.4.47 <a href="#">bd_select_title()</a> . . . . .	70
5.7.4.48 <a href="#">bd_set_player_setting()</a> . . . . .	70
5.7.4.49 <a href="#">bd_set_player_setting_str()</a> . . . . .	71
5.7.4.50 <a href="#">bd_set_rate()</a> . . . . .	71

5.7.4.51 bd_set_scr()	71
5.7.4.52 bd_tell()	72
5.7.4.53 bd_tell_time()	72
5.7.4.54 bd_user_input()	73
5.8 bluray.h	73
5.9 overlay.h File Reference	81
5.9.1 Detailed Description	82
5.9.2 Enumeration Type Documentation	82
5.9.2.1 bd_argb_overlay_cmd_e	82
5.9.2.2 bd_overlay_cmd_e	82
5.9.2.3 bd_overlay_plane_e	83
5.9.3 Function Documentation	83
5.9.3.1 bd_refcnt_inc()	83
5.10 overlay.h	83
5.11 keys.h File Reference	86
5.11.1 Detailed Description	87
5.11.2 Enumeration Type Documentation	87
5.11.2.1 bd_vk_key_e	87
5.12 keys.h	87
5.13 player_settings.h File Reference	88
5.13.1 Detailed Description	90
5.13.2 Enumeration Type Documentation	90
5.13.2.1 anonymous enum	90
5.13.2.2 anonymous enum	91
5.13.2.3 anonymous enum	91
5.13.2.4 anonymous enum	92
5.13.2.5 anonymous enum	92
5.13.2.6 anonymous enum	93
5.13.2.7 anonymous enum	93
5.14 player_settings.h	94
5.15 log_control.h File Reference	95
5.15.1 Detailed Description	96
5.15.2 Typedef Documentation	96
5.15.2.1 BD_LOG_FUNC	96
5.15.3 Enumeration Type Documentation	97
5.15.3.1 debug_mask_t	97
5.15.4 Function Documentation	97
5.15.4.1 bd_get_debug_mask()	97
5.15.4.2 bd_set_debug_handler()	98

5.15.4.3 <code>bd_set_debug_mask()</code> . . . . .	98
5.16 <code>log_control.h</code> . . . . .	98



## Chapter 1

# Deprecated List

Global [bd\\_register\\_dir](#) (BD\_DIR\_OPEN p)

Use [bd\\_open\\_files\(\)](#) instead.

Global [bd\\_register\\_file](#) (BD\_FILE\_OPEN p)

Use [bd\\_open\\_files\(\)](#) instead.



## Chapter 2

# Data Structure Index

### 2.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">BD_ARGB_BUFFER</a>	
Application-allocated frame buffer for ARGB overlays . . . . .	7
<a href="#">BD_ARGB_OVERLAY</a>	
ARGB overlay event . . . . .	8
<a href="#">BD_DIR_H</a>	
Directory access . . . . .	9
<a href="#">BD_DIRENT</a>	
Directory entry . . . . .	10
<a href="#">BD_EVENT</a>	
Event . . . . .	11
<a href="#">BD_FILE_H</a>	
File access . . . . .	11
<a href="#">BD_OVERLAY</a>	
YUV overlay event . . . . .	15
<a href="#">BD_PG_PALETTE_ENTRY</a>	
Overlay palette entry . . . . .	16
<a href="#">BD_PG_RLE_ELEM</a>	
RLE element . . . . .	17
<a href="#">BLURAY_CLIP_INFO</a>	
Clip information . . . . .	17
<a href="#">BLURAY_DISC_INFO</a>	
BluRay disc information . . . . .	18
<a href="#">BLURAY_SOUND_EFFECT</a>	
Sound effect data . . . . .	21
<a href="#">BLURAY_STREAM_INFO</a>	
Clip substream information . . . . .	22
<a href="#">BLURAY_TITLE</a>	
HDMV / BD-J title information . . . . .	23
<a href="#">BLURAY_TITLE_CHAPTER</a>	
Chapter entry . . . . .	24
<a href="#">BLURAY_TITLE_INFO</a>	
Playlist information . . . . .	24

<a href="#">BLURAY_TITLE_MARK</a>	
Playmark information . . . . .	<a href="#">25</a>
<a href="#">META_DL</a>	
DL (Disc Library) metadata entry . . . . .	<a href="#">26</a>
<a href="#">META_THUMBNAIL</a>	
Thumbnail path and resolution . . . . .	<a href="#">27</a>
<a href="#">META_TITLE</a>	
Title name . . . . .	<a href="#">27</a>

## Chapter 3

# File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">filesystem.h</a>	Filesystem interface . . . . .	29
<a href="#">meta_data.h</a>	Disc metadata definitions . . . . .	33
<a href="#">bluray-version.h</a>	Libbluray API version . . . . .	34
<a href="#">bluray.h</a>	Libbluray API . . . . .	35
<a href="#">overlay.h</a>	Graphics overlay events . . . . .	81
<a href="#">keys.h</a>	User input key definitions . . . . .	86
<a href="#">player_settings.h</a>	Definitions for Blu-Ray player settings . . . . .	88
<a href="#">log_control.h</a>	Log control and capture . . . . .	95



## Chapter 4

# Data Structure Documentation

### 4.1 BD\_ARGB\_BUFFER Struct Reference

Application-allocated frame buffer for ARGB overlays.

```
#include <overlay.h>
```

#### Data Fields

- void(\* **lock** )(struct bd\_argb\_buffer\_s \*)  
*Lock (or prepare) buffer for writing.*
- void(\* **unlock** )(struct bd\_argb\_buffer\_s \*)  
*Unlock buffer (write complete)*
- uint32\_t \* **buf** [4]  
*[0] - PG plane, [1] - IG plane.*
- int **width**  
*overlay buffer width (pixels)*
- int **height**  
*overlay buffer height (pixels)*
- struct {  
    uint16\_t **x0**  
        *top-left x coordinate*  
    uint16\_t **y0**  
        *top-left y coordinate*  
    uint16\_t **x1**  
        *bottom-down x coordinate*  
    uint16\_t **y1**  
        *bottom-down y coordinate*  
} **dirty** [2]

*Dirty area of frame buffers.*

### 4.1.1 Detailed Description

Application-allocated frame buffer for ARGB overlays.

When using application-allocated frame buffer DRAW events are executed by libbluray. Application needs to handle only OPEN/FLUSH/CLOSE events.

DRAW events can still be used for optimizations.

### 4.1.2 Field Documentation

#### 4.1.2.1 buf

```
uint32_t* BD_ARGB_BUFFER::buf[4]
```

[0] - PG plane, [1] - IG plane.

[2], [3] reserved for stereoscopic overlay.

#### 4.1.2.2

```
struct { ... } BD_ARGB_BUFFER::dirty[2]
```

Dirty area of frame buffers.

- Updated by library before [lock\(\)](#) call.
- Reset after each BD\_ARGB\_OVERLAY\_FLUSH. [0] - PG plane, [1] - IG plane

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

- [overlay.h](#)

## 4.2 BD\_ARGB\_OVERLAY Struct Reference

ARGB overlay event.

```
#include <overlay.h>
```



## Data Fields

- `int64_t pts`  
*Event timestamp, on video grid.*
- `uint8_t plane`  
*Overlay plane ([bd\\_overlay\\_plane\\_e](#))*
- `uint8_t cmd`  
*Overlay event type ([bd\\_argb\\_overlay\\_cmd\\_e](#))*
- `uint16_t x`  
*top-left x coordinate*
- `uint16_t y`  
*top-left y coordinate*
- `uint16_t w`  
*region width*
- `uint16_t h`  
*region height*
- `uint16_t stride`  
*ARGB buffer stride.*
- `const uint32_t * argb`  
*ARGB image data, 'h' lines, line stride 'stride' pixels.*

### 4.2.1 Detailed Description

ARGB overlay event.

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

- [overlay.h](#)

## 4.3 BD\_DIR\_H Struct Reference

Directory access.

## Data Fields

- `void * internal`  
*reserved for BD\_DIR\_H implementation use*
- `void(* close)(BD_DIR_H *dir)`  
*Close directory stream.*
- `int(* read)(BD_DIR_H *dir, BD\_DIRENT *entry)`  
*Read next directory entry.*

### 4.3.1 Detailed Description

Directory access.

### 4.3.2 Field Documentation

#### 4.3.2.1 close

```
void(* BD_DIR_H::close) (BD_DIR_H *dir)
```

Close directory stream.

##### Parameters

<i>dir</i>	BD_DIR_H object
------------	-----------------

#### 4.3.2.2 read

```
int(* BD_DIR_H::read) (BD_DIR_H *dir, BD\_DIRENT *entry)
```

Read next directory entry.

##### Parameters

<i>dir</i>	BD_DIR_H object
<i>entry</i>	<a href="#">BD_DIRENT</a> where to store directory entry data

##### Returns

0 on success, 1 on EOF, <0 on error

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

- [filesystem.h](#)

## 4.4 BD\_DIRENT Struct Reference

Directory entry.

```
#include <filesystem.h>
```

## Data Fields

- char **d\_name** [256]  
*Null-terminated filename.*

### 4.4.1 Detailed Description

Directory entry.

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

- [filesystem.h](#)

## 4.5 BD\_EVENT Struct Reference

Event.

```
#include <bluray.h>
```

## Data Fields

- uint32\_t **event**  
*Event type ([bd\\_event\\_e](#))*
- uint32\_t **param**  
*Event data.*

### 4.5.1 Detailed Description

Event.

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

- [bluray.h](#)

## 4.6 BD\_FILE\_H Struct Reference

File access.

## Data Fields

- void \* [internal](#)  
*Reserved for BD\_FILE\_H implementation use.*
- void(\* [close](#))(BD\_FILE\_H \*file)  
*Close file.*
- int64\_t(\* [seek](#))(BD\_FILE\_H \*file, int64\_t offset, int32\_t origin)  
*Reposition file offset.*
- int64\_t(\* [tell](#))(BD\_FILE\_H \*file)  
*Get current read or write position.*
- int(\* [eof](#))(BD\_FILE\_H \*file)  
*Check for end of file.*
- int64\_t(\* [read](#))(BD\_FILE\_H \*file, uint8\_t \*buf, int64\_t size)  
*Read from file.*
- int64\_t(\* [write](#))(BD\_FILE\_H \*file, const uint8\_t \*buf, int64\_t size)  
*Write to file.*

### 4.6.1 Detailed Description

File access.

### 4.6.2 Field Documentation

#### 4.6.2.1 close

```
void(* BD_FILE_H::close) (BD_FILE_H *file)
```

Close file.

Parameters

<i>file</i>	BD_FILE_H object
-------------	------------------

#### 4.6.2.2 eof

```
int(* BD_FILE_H::eof) (BD_FILE_H *file)
```

Check for end of file.

- optional, currently not used

**Parameters**

<i>file</i>	BD_FILE_H object
-------------	------------------

**Returns**

1 on EOF, < 0 on error, 0 if not EOF

**4.6.2.3 internal**

```
void* BD_FILE_H::internal
```

Reserved for BD\_FILE\_H implementation use.

Implementation can store here ex. file handle, FILE\*, ...

**4.6.2.4 read**

```
int64_t(* BD_FILE_H::read) (BD_FILE_H *file, uint8_t *buf, int64_t size)
```

Read from file.

**Parameters**

<i>file</i>	BD_FILE_H object
<i>buf</i>	buffer where to store the data
<i>size</i>	bytes to read

**Returns**

number of bytes read, 0 on EOF, < 0 on error

**4.6.2.5 seek**

```
int64_t(* BD_FILE_H::seek) (BD_FILE_H *file, int64_t offset, int32_t origin)
```

Reposition file offset.

- SEEK\_SET: seek to 'offset' bytes from file start
- SEEK\_CUR: seek 'offset' bytes from current position
- SEEK\_END: seek 'offset' bytes from file end

**Parameters**

<i>file</i>	BD_FILE_H object
<i>offset</i>	byte offset
<i>origin</i>	SEEK_SET, SEEK_CUR or SEEK_END

**Returns**

current file offset, < 0 on error

**4.6.2.6 tell**

```
int64_t(* BD_FILE_H::tell) (BD_FILE_H *file)
```

Get current read or write position.

**Parameters**

<i>file</i>	BD_FILE_H object
-------------	------------------

**Returns**

current file offset, < 0 on error

**4.6.2.7 write**

```
int64_t(* BD_FILE_H::write) (BD_FILE_H *file, const uint8_t *buf, int64_t size)
```

Write to file.

Writing 0 bytes can be used to flush previous writes and check for errors.

**Parameters**

<i>file</i>	BD_FILE_H object
<i>buf</i>	data to be written
<i>size</i>	bytes to write

### Returns

number of bytes written, < 0 on error

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

- [filesystem.h](#)

## 4.7 BD\_OVERLAY Struct Reference

YUV overlay event.

```
#include <overlay.h>
```

### Data Fields

- **int64\_t pts**  
*Timestamp, on video grid.*
- **uint8\_t plane**  
*Overlay plane ([bd\\_overlay\\_plane\\_e](#))*
- **uint8\_t cmd**  
*Overlay event type ([bd\\_overlay\\_cmd\\_e](#))*
- **uint8\_t palette\_update\_flag**  
*Set if only overlay palette is changed.*
- **uint16\_t x**  
*top-left x coordinate*
- **uint16\_t y**  
*top-left y coordinate*
- **uint16\_t w**  
*region width*
- **uint16\_t h**  
*region height*
- **const [BD\\_PG\\_PALETTE\\_ENTRY](#) \* palette**  
*overlay palette (256 entries)*
- **const [BD\\_PG\\_RLE\\_ELEM](#) \* img**  
*RLE-compressed overlay image.*

### 4.7.1 Detailed Description

YUV overlay event.

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

- [overlay.h](#)

## 4.8 BD\_PG\_PALETTE\_ENTRY Struct Reference

Overlay palette entry.

```
#include <overlay.h>
```

### Data Fields

- `uint8_t Y`  
*Y component (16...235)*
- `uint8_t Cr`  
*Cr component (16...240)*
- `uint8_t Cb`  
*Cb component (16...240)*
- `uint8_t T`  
*Transparency ( 0...255).*

### 4.8.1 Detailed Description

Overlay palette entry.

Y, Cr and Cb have the same color matrix as the associated video stream.

Entry 0xff is always transparent.

### 4.8.2 Field Documentation

#### 4.8.2.1 T

```
uint8_t BD_PG_PALETTE_ENTRY::T
```

Transparency ( 0...255).

0 - transparent, 255 - opaque.

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

- [overlay.h](#)



## 4.9 BD\_PG\_RLE\_ELEM Struct Reference

RLE element.

```
#include <overlay.h>
```

### Data Fields

- **uint16\_t len**  
*RLE run length.*
- **uint16\_t color**  
*palette index*

### 4.9.1 Detailed Description

RLE element.

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

- [overlay.h](#)

## 4.10 BLURAY\_CLIP\_INFO Struct Reference

Clip information.

```
#include <bluray.h>
```

### Data Fields

- **uint32\_t pkt\_count**  
*Number of mpeg-ts packets.*
- **uint8\_t still\_mode**  
*Clip still mode ([bd\\_still\\_mode\\_e](#))*
- **uint16\_t still\_time**  
*Still time (seconds) if still\_mode == BD\_STILL\_TIME.*
- **uint8\_t video\_stream\_count**  
*Number of video streams.*
- **uint8\_t audio\_stream\_count**  
*Number of audio streams.*
- **uint8\_t pg\_stream\_count**  
*Number of PG (Presentation Graphics) streams.*
- **uint8\_t ig\_stream\_count**  
*Number of IG (Interactive Graphics) streams.*

- `uint8_t sec_audio_stream_count`  
*Number of secondary audio streams.*
- `uint8_t sec_video_stream_count`  
*Number of secondary video streams.*
- `BLURAY_STREAM_INFO * video_streams`  
*Video streams information.*
- `BLURAY_STREAM_INFO * audio_streams`  
*Audio streams information.*
- `BLURAY_STREAM_INFO * pg_streams`  
*PG (Presentation Graphics) streams information.*
- `BLURAY_STREAM_INFO * ig_streams`  
*IG (Interactive Graphics) streams information.*
- `BLURAY_STREAM_INFO * sec_audio_streams`  
*Secondary audio streams information.*
- `BLURAY_STREAM_INFO * sec_video_streams`  
*Secondary video streams information.*
- `uint64_t start_time`  
*start media time, 90kHz, ("playlist time")*
- `uint64_t in_time`  
*start timestamp, 90kHz*
- `uint64_t out_time`  
*end timestamp, 90kHz*
- `char clip_id [6]`  
*Clip identifier (.m2ts file name)*

#### 4.10.1 Detailed Description

Clip information.

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

- [bluray.h](#)

### 4.11 BLURAY\_DISC\_INFO Struct Reference

BluRay disc information.

```
#include <bluray.h>
```

## Data Fields

- uint8\_t **bluray\_detected**  
*1 if BluRay disc was detected*
- const char \* **disc\_name**  
*optional disc name in preferred language*
- const char \* **udf\_volume\_id**  
*optional UDF volume identifier*
- uint8\_t **disc\_id** [20]  
*Disc ID.*
- uint8\_t **no\_menu\_support**  
*HDMV / BD-J titles.*
- uint8\_t **first\_play\_supported**  
*1 if First Play title is present on the disc and can be played*
- uint8\_t **top\_menu\_supported**  
*1 if Top Menu title is present on the disc and can be played*
- uint32\_t **num\_titles**  
*number of titles on the disc, not including "First Play" and "Top Menu"*
- const BLURAY\_TITLE \* **titles**  
*index is title number 1 ... N*
- const BLURAY\_TITLE \* **first\_play**  
*titles[N+1].*
- const BLURAY\_TITLE \* **top\_menu**  
*titles[0].*
- uint32\_t **num\_hdmv\_titles**  
*number of HDMV titles*
- uint32\_t **num\_bdj\_titles**  
*number of BD-J titles*
- uint32\_t **num\_unsupported\_titles**  
*number of unsupported titles*
- uint8\_t **bdj\_detected**  
*BD-J info (valid only if disc uses BD-J)*
- uint8\_t **bdj\_supported**  
*(deprecated)*
- uint8\_t **libjvm\_detected**  
*1 if usable Java VM was found*
- uint8\_t **bdj\_handled**  
*1 if usable Java VM + libbluray.jar was found*
- char **bdj\_org\_id** [9]  
*(BD-J) disc organization ID*
- char **bdj\_disc\_id** [33]  
*(BD-J) disc ID*
- uint8\_t **video\_format**  
*bd\_video\_format\_e*
- uint8\_t **frame\_rate**  
*bd\_video\_rate\_e*
- uint8\_t **content\_exist\_3D**

- 1 if 3D content exists on the disc*
- uint8\_t **initial\_output\_mode\_preference**
  - 0 - 2D, 1 - 3D*
- uint8\_t **provider\_data** [32]
  - Content provider data.*
- uint8\_t **aacs\_detected**
  - 1 if disc is using AACs encoding*
- uint8\_t **libaacs\_detected**
  - 1 if usable AACs decoding library was found*
- uint8\_t **aacs\_handled**
  - 1 if disc is using supported AACs encoding*
- int **aacs\_error\_code**
  - AACs error code (BD\_AACS\_\*)*
- int **aacs\_mkbv**
  - AACs MKB version.*
- uint8\_t **bdplus\_detected**
  - 1 if disc is using BD+ encoding*
- uint8\_t **libbdplus\_detected**
  - 1 if usable BD+ decoding library was found*
- uint8\_t **bdplus\_handled**
  - 1 if disc is using supported BD+ encoding*
- uint8\_t **bdplus\_gen**
  - BD+ content code generation.*
- uint32\_t **bdplus\_date**
  - BD+ content code release date ((year<<16)|(month<<8)|day)*
- uint8\_t **initial\_dynamic\_range\_type**
  - bd\_dynamic\_range\_type\_e*

### 4.11.1 Detailed Description

BluRay disc information.

### 4.11.2 Field Documentation

#### 4.11.2.1 bdj\_detected

```
uint8_t BLURAY_DISC_INFO::bdj_detected
```

BD-J info (valid only if disc uses BD-J)

1 if disc uses BD-J

#### 4.11.2.2 first\_play

```
const BLURAY_TITLE* BLURAY_DISC_INFO::first_play
```

titles[N+1].

NULL if not present on the disc.

#### 4.11.2.3 no\_menu\_support

```
uint8_t BLURAY_DISC_INFO::no_menu_support
```

HDMV / BD-J titles.

1 if this disc can't be played using on-disc menus

#### 4.11.2.4 top\_menu

```
const BLURAY_TITLE* BLURAY_DISC_INFO::top_menu
```

titles[0].

NULL if not present on the disc.

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

- [bluray.h](#)

## 4.12 BLURAY\_SOUND\_EFFECT Struct Reference

Sound effect data.

```
#include <bluray.h>
```

### Data Fields

- **uint8\_t num\_channels**  
*1 - mono, 2 - stereo*
- **uint32\_t num\_frames**  
*Number of audio frames.*
- **const int16\_t \* samples**  
*48000 Hz, 16 bit LPCM.*

### 4.12.1 Detailed Description

Sound effect data.

### 4.12.2 Field Documentation

#### 4.12.2.1 samples

```
const int16_t* BLURAY_SOUND_EFFECT::samples
```

48000 Hz, 16 bit LPCM.

Interleaved if stereo

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

- [bluray.h](#)

## 4.13 BLURAY\_STREAM\_INFO Struct Reference

Clip substream information.

```
#include <bluray.h>
```

### Data Fields

- **uint8\_t coding\_type**  
*Stream coding ([bd\\_stream\\_type\\_e](#))*
- **uint8\_t format**  
*Stream format ([bd\\_video\\_format\\_e](#) or [bd\\_audio\\_format\\_e](#))*
- **uint8\_t rate**  
*Stream frame rate ([bd\\_audio\\_rate\\_e](#) or [bd\\_video\\_rate\\_e](#))*
- **uint8\_t char\_code**  
*Text subtitle character code ([bd\\_char\\_code\\_e](#))*
- **uint8\_t lang** [4]  
*Language code.*
- **uint16\_t pid**  
*mpeg-ts PID*
- **uint8\_t aspect**  
*Stream video aspect ratio ([bd\\_video\\_aspect\\_e](#))*
- **uint8\_t subpath\_id**  
*Sub path identifier (= separate mpeg-ts mux / .m2ts file)*

### 4.13.1 Detailed Description

Clip substream information.

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

- [bluray.h](#)

## 4.14 BLURAY\_TITLE Struct Reference

HDMV / BD-J title information.

```
#include <bluray.h>
```

### Data Fields

- `const char * name`  
*optional title name in preferred language*
- `uint8_t interactive`  
*1 if title is interactive (title length and playback position should not be shown in UI)*
- `uint8_t accessible`  
*1 if it is allowed to jump into this title*
- `uint8_t hidden`  
*1 if title number should not be shown during playback*
- `uint8_t bdj`  
*0 - HDMV title.*
- `uint32_t id_ref`  
*Movie Object number / bdjo file number.*

### 4.14.1 Detailed Description

HDMV / BD-J title information.

### 4.14.2 Field Documentation

#### 4.14.2.1 bdj

```
uint8_t BLURAY_TITLE::bdj
```

0 - HDMV title.

1 - BD-J title

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

- [bluray.h](#)

### 4.15 BLURAY\_TITLE\_CHAPTER Struct Reference

Chapter entry.

```
#include <bluray.h>
```

#### Data Fields

- uint32\_t **idx**  
*Chapter index (number - 1)*
- uint64\_t **start**  
*start media time, 90kHz, ("playlist time")*
- uint64\_t **duration**  
*duration*
- uint64\_t **offset**  
*distance from title start, bytes*
- unsigned **clip\_ref**  
*Clip reference (index to playlist clips list)*

#### 4.15.1 Detailed Description

Chapter entry.

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

- [bluray.h](#)

### 4.16 BLURAY\_TITLE\_INFO Struct Reference

Playlist information.

```
#include <bluray.h>
```



## Data Fields

- `uint32_t idx`  
*Playlist index number (filled only with [bd\\_get\\_title\\_info\(\)](#))*
- `uint32_t playlist`  
*Playlist ID (mpls file name)*
- `uint64_t duration`  
*Playlist duration, 90 kHz.*
- `uint32_t clip_count`  
*Number of clips.*
- `uint8_t angle_count`  
*Number of angles.*
- `uint32_t chapter_count`  
*Number of chapters.*
- `uint32_t mark_count`  
*Number of playmarks.*
- [BLURAY\\_CLIP\\_INFO](#) \* `clips`  
*Clip information.*
- [BLURAY\\_TITLE\\_CHAPTER](#) \* `chapters`  
*Chapter information.*
- [BLURAY\\_TITLE\\_MARK](#) \* `marks`  
*Playmark information.*
- `uint8_t mvc_base_view_r_flag`  
*MVC base view (0 - left, 1 - right)*

### 4.16.1 Detailed Description

Playlist information.

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

- [bluray.h](#)

## 4.17 BLURAY\_TITLE\_MARK Struct Reference

Playmark information.

```
#include <bluray.h>
```

## Data Fields

- uint32\_t **idx**  
*Mark index (number - 1)*
- int **type**  
[bd\\_mark\\_type\\_e](#)
- uint64\_t **start**  
*mark media time, 90kHz, ("playlist time")*
- uint64\_t **duration**  
*time to next mark*
- uint64\_t **offset**  
*mark distance from title start, bytes*
- unsigned **clip\_ref**  
*Clip reference (index to playlist clips list)*

### 4.17.1 Detailed Description

Playmark information.

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

- [bluray.h](#)

## 4.18 META\_DL Struct Reference

DL (Disc Library) metadata entry.

```
#include <meta_data.h>
```

## Data Fields

- char **language\_code** [4]  
*Language used in this metadata entry.*
- char \* **filename**  
*Source file (relative to disc root)*
- char \* **di\_name**  
*Disc name.*
- char \* **di\_alternative**  
*Alternative name.*
- uint8\_t **di\_num\_sets**  
*Number of discs in original volume or collection.*
- uint8\_t **di\_set\_number**  
*Sequence order of the disc from an original collection.*
- uint32\_t **toc\_count**  
*Number of title entries.*
- [META\\_TITLE](#) \* **toc\_entries**  
*Title data.*
- uint8\_t **thumb\_count**  
*Number of thumbnails.*
- [META\\_THUMBNAIL](#) \* **thumbnails**  
*Thumbnail data.*

### 4.18.1 Detailed Description

DL (Disc Library) metadata entry.

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

- [meta\\_data.h](#)

## 4.19 META\_THUMBNAIL Struct Reference

Thumbnail path and resolution.

```
#include <meta_data.h>
```

### Data Fields

- `char * path`  
*Path to thumbnail image (relative to disc root)*
- `uint32_t xres`  
*Thumbnail width.*
- `uint32_t yres`  
*Thumbnail height.*

### 4.19.1 Detailed Description

Thumbnail path and resolution.

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

- [meta\\_data.h](#)

## 4.20 META\_TITLE Struct Reference

Title name.

```
#include <meta_data.h>
```

### Data Fields

- `uint32_t title_number`  
*Title number (from disc index)*
- `char * title_name`  
*Title name.*

### 4.20.1 Detailed Description

Title name.

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

- [meta\\_data.h](#)



## Chapter 5

# File Documentation

### 5.1 filesystem.h File Reference

Filesystem interface.

```
#include <stdint.h>
```

#### Data Structures

- struct [BD\\_FILE\\_H](#)  
*File access.*
- struct [BD\\_DIRENT](#)  
*Directory entry.*
- struct [BD\\_DIR\\_H](#)  
*Directory access.*

#### Typedefs

- typedef [BD\\_FILE\\_H](#) [\\*\(\\* BD\\_FILE\\_OPEN\)](#) (const char \*filename, const char \*mode)  
*Open a file.*
- typedef [BD\\_DIR\\_H](#) [\\*\(\\* BD\\_DIR\\_OPEN\)](#) (const char \*dirname)  
*Open a directory.*

#### Functions

- [BD\\_FILE\\_OPEN](#) [bd\\_register\\_file](#) ([BD\\_FILE\\_OPEN](#) p)  
*Register function pointer that will be used to open a file.*
- [BD\\_DIR\\_OPEN](#) [bd\\_register\\_dir](#) ([BD\\_DIR\\_OPEN](#) p)  
*Register function pointer that will be used to open a directory.*

### 5.1.1 Detailed Description

Filesystem interface.

File access wrappers can be used to bind libbluray to external filesystem. Typical use case would be playing BluRay from network filesystem.

### 5.1.2 Typedef Documentation

#### 5.1.2.1 BD\_DIR\_OPEN

```
typedef BD_DIR_H *(* BD_DIR_OPEN) (const char *dirname)
```

Open a directory.

Prototype for a function that returns BD\_DIR\_H implementation.

##### Parameters

<i>dirname</i>	name of the directory to open
----------------	-------------------------------

##### Returns

BD\_DIR\_H object, NULL on error

#### 5.1.2.2 BD\_FILE\_OPEN

```
typedef BD_FILE_H *(* BD_FILE_OPEN) (const char *filename, const char *mode)
```

Open a file.

Prototype for a function that returns BD\_FILE\_H implementation.

##### Parameters

<i>filename</i>	name of the file to open
<i>mode</i>	string starting with "r" for reading or "w" for writing

#### Returns

BD\_FILE\_H object, NULL on error

### 5.1.3 Function Documentation

#### 5.1.3.1 bd\_register\_dir()

```
BD_DIR_OPEN bd_register_dir (  
    BD_DIR_OPEN p )
```

Register function pointer that will be used to open a directory.

**Deprecated** Use `bd_open_files()` instead.

#### Parameters

<i>p</i>	function pointer
----------	------------------

#### Returns

previous function pointer registered

#### 5.1.3.2 bd\_register\_file()

```
BD_FILE_OPEN bd_register_file (  
    BD_FILE_OPEN p )
```

Register function pointer that will be used to open a file.

**Deprecated** Use `bd_open_files()` instead.

#### Parameters

<i>p</i>	function pointer
----------	------------------

## Returns

previous function pointer registered

## 5.2 filesystem.h

[Go to the documentation of this file.](#)

```

1 /*
2  * This file is part of libbluray
3  * Copyright (C) 2009-2010  Obliter0n
4  * Copyright (C) 2009-2010  John Stebbins
5  *
6  * This library is free software; you can redistribute it and/or
7  * modify it under the terms of the GNU Lesser General Public
8  * License as published by the Free Software Foundation; either
9  * version 2.1 of the License, or (at your option) any later version.
10 *
11 * This library is distributed in the hope that it will be useful,
12 * but WITHOUT ANY WARRANTY; without even the implied warranty of
13 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
14 * Lesser General Public License for more details.
15 *
16 * You should have received a copy of the GNU Lesser General Public
17 * License along with this library. If not, see
18 * <http://www.gnu.org/licenses/>.
19 */
20
21 #ifndef BD_FILESYSTEM_H_
22 #define BD_FILESYSTEM_H_
23
24 #ifdef __cplusplus
25 extern "C" {
26 #endif
27
28 #include <stdint.h>
29
30 typedef struct bd_file_s BD_FILE_H;
31 struct bd_file_s
32 {
33     void* internal;
34
35     void (*close) (BD_FILE_H *file);
36
37     int64_t (*seek) (BD_FILE_H *file, int64_t offset, int32_t origin);
38
39     int64_t (*tell) (BD_FILE_H *file);
40
41     int (*eof) (BD_FILE_H *file);
42
43     int64_t (*read) (BD_FILE_H *file, uint8_t *buf, int64_t size);
44
45     int64_t (*write) (BD_FILE_H *file, const uint8_t *buf, int64_t size);
46 };
47
48 typedef struct
49 {
50     char d_name[256];
51 } BD_DIRENT;
52
53 typedef struct bd_dir_s BD_DIR_H;
54 struct bd_dir_s
55 {
56     void* internal;
57
58     void (*close) (BD_DIR_H *dir);
59
60     int (*read) (BD_DIR_H *dir, BD_DIRENT *entry);
61 };
62
63 typedef BD_FILE_H* (*BD_FILE_OPEN) (const char* filename, const char *mode);
64
65 typedef BD_DIR_H* (*BD_DIR_OPEN) (const char* dirname);
66
67 BD_FILE_OPEN bd_register_file(BD_FILE_OPEN p);
68
69 BD_DIR_OPEN bd_register_dir(BD_DIR_OPEN p);

```



```

186
187 #ifdef __cplusplus
188 }
189 #endif
190
191 #endif /* BD_FILESYSTEM_H_ */

```

## 5.3 meta\_data.h File Reference

Disc metadata definitions.

```
#include <stdint.h>
```

### Data Structures

- struct [META\\_THUMBNAIL](#)  
*Thumbnail path and resolution.*
- struct [META\\_TITLE](#)  
*Title name.*
- struct [META\\_DL](#)  
*DL (Disc Library) metadata entry.*

### 5.3.1 Detailed Description

Disc metadata definitions.

## 5.4 meta\_data.h

[Go to the documentation of this file.](#)

```

1 /*
2  * This file is part of libbluray
3  * Copyright (C) 2010 fraxinas
4  *
5  * This library is free software; you can redistribute it and/or
6  * modify it under the terms of the GNU Lesser General Public
7  * License as published by the Free Software Foundation; either
8  * version 2.1 of the License, or (at your option) any later version.
9  *
10 * This library is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
13 * Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public
16 * License along with this library. If not, see
17 * <http://www.gnu.org/licenses/>.
18 */
19
25 #if !defined(_META_DATA_H_)
26 #define _META_DATA_H_
27
28 #include <stdint.h>
29
31 typedef struct meta_thumbnail {

```

```

32     char *          path;
33     uint32_t        xres;
34     uint32_t        yres;
35 } META_THUMBNAIL;
36
37 typedef struct meta_title {
38     uint32_t        title_number;
39     char *          title_name;
40 } META_TITLE;
41
42 typedef struct meta_dl {
43     char            language_code[4];
44     char *          filename;
45     char *          di_name;
46     char *          di_alternative;
47     uint8_t         di_num_sets;
48     uint8_t         di_set_number;
49     uint32_t        toc_count;
50     META_TITLE *    toc_entries;
51     uint8_t         thumb_count;
52     META_THUMBNAIL * thumbnails;
53 } META_DL;
54
55 #endif // _META_DATA_H_
56
57
58

```

## 5.5 bluray-version.h File Reference

libbluray API version

### Macros

- #define **BLURAY\_VERSION\_CODE**(major, minor, micro)  
*Pack version number to single integer.*
- #define **BLURAY\_VERSION\_MAJOR** 1  
*libbluray major version number*
- #define **BLURAY\_VERSION\_MINOR** 3  
*libbluray minor version number*
- #define **BLURAY\_VERSION\_MICRO** 4  
*libbluray micro version number*
- #define **BLURAY\_VERSION\_STRING** "1.3.4"  
*libbluray version number as a string*
- #define **BLURAY\_VERSION** **BLURAY\_VERSION\_CODE**(**BLURAY\_VERSION\_MAJOR**, **BLURAY\_VERSION\_MINOR**, **BLURAY\_VERSION\_MICRO**)  
*libbluray version number as a single integer*

### 5.5.1 Detailed Description

libbluray API version

### 5.5.2 Macro Definition Documentation

### 5.5.2.1 BLURAY\_VERSION\_CODE

```
#define BLURAY_VERSION_CODE(
    major,
    minor,
    micro )
```

**Value:**

```
((major) * 10000) +
(minor) * 100) +
(micro) * 1))
```

Pack version number to single integer.

## 5.6 bluray-version.h

[Go to the documentation of this file.](#)

```
1 /*
2  * This file is part of libbluray
3  * Copyright (C) 2011 hpil
4  *
5  * This library is free software; you can redistribute it and/or
6  * modify it under the terms of the GNU Lesser General Public
7  * License as published by the Free Software Foundation; either
8  * version 2.1 of the License, or (at your option) any later version.
9  *
10 * This library is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
13 * Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public
16 * License along with this library. If not, see
17 * <http://www.gnu.org/licenses/>.
18 */
19
20 #ifndef BLURAY_VERSION_H_
21 #define BLURAY_VERSION_H_
22
23 #define BLURAY_VERSION_CODE(major, minor, micro) \
24     ((major) * 10000) + \
25     ((minor) * 100) + \
26     (micro)
27
28 #define BLURAY_VERSION_MAJOR 1
29 #define BLURAY_VERSION_MINOR 3
30 #define BLURAY_VERSION_MICRO 4
31
32 #define BLURAY_VERSION_STRING "1.3.4"
33
34 #define BLURAY_VERSION \
35     BLURAY_VERSION_CODE(BLURAY_VERSION_MAJOR, BLURAY_VERSION_MINOR, BLURAY_VERSION_MICRO)
36
37 #endif /* BLURAY_VERSION_H_ */
```

## 5.7 bluray.h File Reference

libbluray API

```
#include <stdint.h>
```

## Data Structures

- struct [BLURAY\\_TITLE](#)  
*HDMV / BD-J title information.*
- struct [BLURAY\\_DISC\\_INFO](#)  
*BluRay disc information.*
- struct [BLURAY\\_STREAM\\_INFO](#)  
*Clip substream information.*
- struct [BLURAY\\_CLIP\\_INFO](#)  
*Clip information.*
- struct [BLURAY\\_TITLE\\_CHAPTER](#)  
*Chapter entry.*
- struct [BLURAY\\_TITLE\\_MARK](#)  
*Playmark information.*
- struct [BLURAY\\_TITLE\\_INFO](#)  
*Playlist information.*
- struct [BLURAY\\_SOUND\\_EFFECT](#)  
*Sound effect data.*
- struct [BD\\_EVENT](#)  
*Event.*

## Macros

- `#define TITLES_ALL 0`  
*all titles.*
- `#define TITLES_FILTER_DUP_TITLE 0x01`  
*remove duplicate titles.*
- `#define TITLES_FILTER_DUP_CLIP 0x02`  
*remove titles that have duplicate clips.*
- `#define TITLES_RELEVANT (TITLES_FILTER_DUP_TITLE | TITLES_FILTER_DUP_CLIP)`  
*remove duplicate titles and clips*
- `#define BD_AACS_CORRUPTED_DISC -1`  
*Corrupt disc (missing/invalid files)*
- `#define BD_AACS_NO_CONFIG -2`  
*AACS configuration file missing*
- `#define BD_AACS_NO_PK -3`  
*No valid processing key found*
- `#define BD_AACS_NO_CERT -4`  
*No valid certificate found*
- `#define BD_AACS_CERT_REVOKED -5`  
*All certificates have been revoked*
- `#define BD_AACS_MMC_FAILED -6`  
*MMC (disc drive interaction) failed*

- **#define BLURAY\_AUDIO\_STREAM 0**  
*Select audio stream*
- **#define BLURAY\_PG\_TEXTST\_STREAM 1**  
*Select subtitle stream*
- **#define BD\_ERROR\_HDMV 1**  
*HDMV VM failed to play the title*
- **#define BD\_ERROR\_BDJ 2**  
*BD-J failed to play the title*
- **#define BD\_ERROR\_AACS 3**  
*AACS failed or not supported*
- **#define BD\_ERROR\_BDPLUS 4**  
*BD+ failed or not supported*
- **#define BLURAY\_TITLE\_FIRST\_PLAY 0xffff**  
*"First Play" title started*
- **#define BLURAY\_TITLE\_TOP\_MENU 0**  
*"Top Menu" title started*
- **#define BLURAY\_KIT\_PLAY 0x1**  
*BD-J requests to handle "Play" UO*
- **#define BLURAY\_KIT\_STOP 0x2**  
*BD-J requests to handle "Stop" UO*
- **#define BLURAY\_KIT\_FFW 0x4**  
*BD-J requests to handle "Fast Forward" UO*
- **#define BLURAY\_KIT\_REW 0x8**  
*BD-J requests to handle "Reverse" UO*
- **#define BLURAY\_KIT\_TRACK\_NEXT 0x10**  
*BD-J requests to handle "Next Track" UO*
- **#define BLURAY\_KIT\_TRACK\_PREV 0x20**  
*BD-J requests to handle "Prev Track" UO*
- **#define BLURAY\_KIT\_PAUSE 0x40**  
*BD-J requests to handle "Pause" UO*
- **#define BLURAY\_KIT\_STILL\_OFF 0x80**  
*BD-J requests to handle "Still Off" UO*
- **#define BLURAY\_KIT\_SEC\_AUDIO 0x100**  
*BD-J requests to handle "Sec. Audio" UO*

- `#define BLURAY_KIT_SEC_VIDEO 0x200`  
*BD-J requests to handle "Sec. Video" UO*
- `#define BLURAY_KIT_PG_TEXTST 0x400`  
*BD-J requests to handle "Subtitle" UO*
- `#define BLURAY_UO_MENU_CALL 0x1`  
*"Menu Call" masked (not allowed)*
- `#define BLURAY_UO_TITLE_SEARCH 0x2`  
*"Title Search" masked (not allowed)*
- `#define BLURAY_RATE_PAUSED 0`  
*Set playback rate to PAUSED*
- `#define BLURAY_RATE_NORMAL 90000`  
*Set playback rate to NORMAL*

## Typedefs

- typedef struct bluray [BLURAY](#)  
*This structure is opaque.*
- typedef void(\* [bd\\_overlay\\_proc\\_f](#)) (void \*handle, const struct bd\_overlay\_s \*const event)  
*YUV overlay handler function type.*
- typedef void(\* [bd\\_argb\\_overlay\\_proc\\_f](#)) (void \*handle, const struct bd\_argb\_overlay\_s \*const event)  
*ARGB overlay handler function type.*

## Enumerations

- enum [bd\\_stream\\_type\\_e](#) {  
**BLURAY\_STREAM\_TYPE\_VIDEO\_MPEG1** = 0x01 ,  
**BLURAY\_STREAM\_TYPE\_VIDEO\_MPEG2** = 0x02 ,  
**BLURAY\_STREAM\_TYPE\_AUDIO\_MPEG1** = 0x03 ,  
**BLURAY\_STREAM\_TYPE\_AUDIO\_MPEG2** = 0x04 ,  
**BLURAY\_STREAM\_TYPE\_AUDIO\_LPCM** = 0x80 ,  
**BLURAY\_STREAM\_TYPE\_AUDIO\_AC3** = 0x81 ,  
**BLURAY\_STREAM\_TYPE\_AUDIO\_DTS** = 0x82 ,  
**BLURAY\_STREAM\_TYPE\_AUDIO\_TRUHD** = 0x83 ,  
**BLURAY\_STREAM\_TYPE\_AUDIO\_AC3PLUS** = 0x84 ,  
**BLURAY\_STREAM\_TYPE\_AUDIO\_DTSHD** = 0x85 ,  
**BLURAY\_STREAM\_TYPE\_AUDIO\_DTSHD\_MASTER** = 0x86 ,  
**BLURAY\_STREAM\_TYPE\_VIDEO\_VC1** = 0xea ,  
**BLURAY\_STREAM\_TYPE\_VIDEO\_H264** = 0x1b ,  
**BLURAY\_STREAM\_TYPE\_VIDEO\_HEVC** = 0x24 ,  
**BLURAY\_STREAM\_TYPE\_SUB\_PG** = 0x90 ,  
**BLURAY\_STREAM\_TYPE\_SUB\_IG** = 0x91 ,  
**BLURAY\_STREAM\_TYPE\_SUB\_TEXT** = 0x92 ,  
**BLURAY\_STREAM\_TYPE\_AUDIO\_AC3PLUS\_SECONDARY** = 0xa1 ,  
**BLURAY\_STREAM\_TYPE\_AUDIO\_DTSHD\_SECONDARY** = 0xa2 }

*Stream video coding type.*

- enum `bd_video_format_e` {  
`BLURAY_VIDEO_FORMAT_480I` = 1 ,  
`BLURAY_VIDEO_FORMAT_576I` = 2 ,  
`BLURAY_VIDEO_FORMAT_480P` = 3 ,  
`BLURAY_VIDEO_FORMAT_1080I` = 4 ,  
`BLURAY_VIDEO_FORMAT_720P` = 5 ,  
`BLURAY_VIDEO_FORMAT_1080P` = 6 ,  
`BLURAY_VIDEO_FORMAT_576P` = 7 ,  
`BLURAY_VIDEO_FORMAT_2160P` = 8 }

*Stream video format.*

- enum `bd_video_rate_e` {  
`BLURAY_VIDEO_RATE_24000_1001` = 1 ,  
`BLURAY_VIDEO_RATE_24` = 2 ,  
`BLURAY_VIDEO_RATE_25` = 3 ,  
`BLURAY_VIDEO_RATE_30000_1001` = 4 ,  
`BLURAY_VIDEO_RATE_50` = 6 ,  
`BLURAY_VIDEO_RATE_60000_1001` = 7 }

*Stream video frame rate.*

- enum `bd_video_aspect_e` {  
`BLURAY_ASPECT_RATIO_4_3` = 2 ,  
`BLURAY_ASPECT_RATIO_16_9` = 3 }

*Stream video aspect ratio.*

- enum `bd_audio_format_e` {  
`BLURAY_AUDIO_FORMAT_MONO` = 1 ,  
`BLURAY_AUDIO_FORMAT_STEREO` = 3 ,  
`BLURAY_AUDIO_FORMAT_MULTI_CHAN` = 6 ,  
`BLURAY_AUDIO_FORMAT_COMBO` = 12 }

*Stream audio format.*

- enum `bd_audio_rate_e` {  
`BLURAY_AUDIO_RATE_48` = 1 ,  
`BLURAY_AUDIO_RATE_96` = 4 ,  
`BLURAY_AUDIO_RATE_192` = 5 ,  
`BLURAY_AUDIO_RATE_192_COMBO` = 12 ,  
`BLURAY_AUDIO_RATE_96_COMBO` = 14 }

*Stream audio rate.*

- enum `bd_char_code_e` {  
`BLURAY_TEXT_CHAR_CODE_UTF8` = 0x01 ,  
`BLURAY_TEXT_CHAR_CODE_UTF16BE` = 0x02 ,  
`BLURAY_TEXT_CHAR_CODE_SHIFT_JIS` = 0x03 ,  
`BLURAY_TEXT_CHAR_CODE_EUC_KR` = 0x04 ,  
`BLURAY_TEXT_CHAR_CODE_GB18030_20001` = 0x05 ,  
`BLURAY_TEXT_CHAR_CODE_CN_GB` = 0x06 ,  
`BLURAY_TEXT_CHAR_CODE_BIG5` = 0x07 }

*Text subtitle charset.*

- enum `bd_still_mode_e` {  
`BLURAY_STILL_NONE` = 0x00 ,  
`BLURAY_STILL_TIME` = 0x01 ,  
`BLURAY_STILL_INFINITE` = 0x02 }

*Clip still mode type.*

- enum `bd_mark_type_e` {  
`BLURAY_MARK_ENTRY` = 0x01 ,  
`BLURAY_MARK_LINK` = 0x02 }

*Mark type.*

- enum `bd_dynamic_range_type_e` {  
**BLURAY\_DYNAMIC\_RANGE\_SDR** = 0 ,  
**BLURAY\_DYNAMIC\_RANGE\_HDR10** = 1 ,  
**BLURAY\_DYNAMIC\_RANGE\_DOLBY\_VISION** = 2 }

*Clip dynamic range.*

- enum `bd_player_setting` {  
**BLURAY\_PLAYER\_SETTING\_AUDIO\_LANG** = 16 ,  
**BLURAY\_PLAYER\_SETTING\_PG\_LANG** = 17 ,  
**BLURAY\_PLAYER\_SETTING\_MENU\_LANG** = 18 ,  
**BLURAY\_PLAYER\_SETTING\_COUNTRY\_CODE** = 19 ,  
**BLURAY\_PLAYER\_SETTING\_REGION\_CODE** = 20 ,  
**BLURAY\_PLAYER\_SETTING\_OUTPUT\_PREFER** = 21 ,  
**BLURAY\_PLAYER\_SETTING\_PARENTAL** = 13 ,  
**BLURAY\_PLAYER\_SETTING\_AUDIO\_CAP** = 15 ,  
**BLURAY\_PLAYER\_SETTING\_VIDEO\_CAP** = 29 ,  
**BLURAY\_PLAYER\_SETTING\_DISPLAY\_CAP** = 23 ,  
**BLURAY\_PLAYER\_SETTING\_3D\_CAP** = 24 ,  
**BLURAY\_PLAYER\_SETTING\_UHD\_CAP** = 25 ,  
**BLURAY\_PLAYER\_SETTING\_UHD\_DISPLAY\_CAP** = 26 ,  
**BLURAY\_PLAYER\_SETTING\_HDR\_PREFERENCE** = 27 ,  
**BLURAY\_PLAYER\_SETTING\_SDR\_CONV\_PREFER** = 28 ,  
**BLURAY\_PLAYER\_SETTING\_TEXT\_CAP** = 30 ,  
**BLURAY\_PLAYER\_SETTING\_PLAYER\_PROFILE** = 31 ,  
**BLURAY\_PLAYER\_SETTING\_DECODE\_PG** = 0x100 ,  
**BLURAY\_PLAYER\_SETTING\_PERSISTENT\_STORAGE** = 0x101 ,  
**BLURAY\_PLAYER\_PERSISTENT\_ROOT** = 0x200 ,  
**BLURAY\_PLAYER\_CACHE\_ROOT** = 0x201 ,  
**BLURAY\_PLAYER\_JAVA\_HOME** = 0x202 }

*Player setting.*

- enum `bd_event_e` {  
**BD\_EVENT\_NONE** = 0 ,  
**BD\_EVENT\_ERROR** = 1 ,  
**BD\_EVENT\_READ\_ERROR** = 2 ,  
**BD\_EVENT\_ENCRYPTED** = 3 ,  
**BD\_EVENT\_ANGLE** = 4 ,  
**BD\_EVENT\_TITLE** = 5 ,  
**BD\_EVENT\_PLAYLIST** = 6 ,  
**BD\_EVENT\_PLAYITEM** = 7 ,  
**BD\_EVENT\_CHAPTER** = 8 ,  
**BD\_EVENT\_PLAYMARK** = 9 ,  
**BD\_EVENT\_END\_OF\_TITLE** = 10 ,  
**BD\_EVENT\_AUDIO\_STREAM** = 11 ,  
**BD\_EVENT\_IG\_STREAM** = 12 ,  
**BD\_EVENT\_PG\_TEXTST\_STREAM** = 13 ,  
**BD\_EVENT\_PIP\_PG\_TEXTST\_STREAM** = 14 ,  
**BD\_EVENT\_SECONDARY\_AUDIO\_STREAM** = 15 ,  
**BD\_EVENT\_SECONDARY\_VIDEO\_STREAM** = 16 ,  
**BD\_EVENT\_PG\_TEXTST** = 17 ,  
**BD\_EVENT\_PIP\_PG\_TEXTST** = 18 ,  
**BD\_EVENT\_SECONDARY\_AUDIO** = 19 ,  
**BD\_EVENT\_SECONDARY\_VIDEO** = 20 ,  
**BD\_EVENT\_SECONDARY\_VIDEO\_SIZE** = 21 ,  
**BD\_EVENT\_PLAYLIST\_STOP** = 22 ,



```

BD_EVENT_DISCONTINUITY = 23 ,
BD_EVENT_SEEK = 24 ,
BD_EVENT_STILL = 25 ,
BD_EVENT_STILL_TIME = 26 ,
BD_EVENT_SOUND_EFFECT = 27 ,
BD_EVENT_IDLE = 28 ,
BD_EVENT_POPUP = 29 ,
BD_EVENT_MENU = 30 ,
BD_EVENT_STEREOSCOPIC_STATUS = 31 ,
BD_EVENT_KEY_INTEREST_TABLE = 32 ,
BD_EVENT_UO_MASK_CHANGED = 33 }

```

*Event type.*

## Functions

- void `bd_get_version` (int \*major, int \*minor, int \*micro)  
*Get libbluray version.*
- `BLURAY * bd_open` (const char \*device\_path, const char \*keyfile\_path)  
*Open BluRay disc.*
- `BLURAY * bd_init` (void)  
*Initialize BLURAY object.*
- int `bd_open_disc` (`BLURAY *bd`, const char \*device\_path, const char \*keyfile\_path)  
*Open BluRay disc.*
- int `bd_open_stream` (`BLURAY *bd`, void \*read\_blocks\_handle, int(\*read\_blocks)(void \*handle, void \*buf, int lba, int num\_blocks))  
*Open BluRay disc.*
- int `bd_open_files` (`BLURAY *bd`, void \*handle, struct `bd_dir_s` \*(\*open\_dir)(void \*handle, const char \*rel\_path), struct `bd_file_s` \*(\*open\_file)(void \*handle, const char \*rel\_path))  
*Open BluRay disc.*
- void `bd_close` (`BLURAY *bd`)  
*Close BluRay disc.*
- const `BLURAY_DISC_INFO` \* `bd_get_disc_info` (`BLURAY *bd`)  
*Get information about current BluRay disc.*
- const struct meta\_dl \* `bd_get_meta` (`BLURAY *bd`)  
*Get meta information about current BluRay disc.*
- int `bd_get_meta_file` (`BLURAY *bd`, const char \*file\_name, void \*\*data, int64\_t \*size)  
*Read metadata file from BluRay disc.*
- uint32\_t `bd_get_titles` (`BLURAY *bd`, uint8\_t flags, uint32\_t min\_title\_length)  
*Get number of titles (playlists)*
- int `bd_get_main_title` (`BLURAY *bd`)  
*Get main title Returned number is an index to the list created by `bd_get_titles()`*
- `BLURAY_TITLE_INFO` \* `bd_get_title_info` (`BLURAY *bd`, uint32\_t title\_idx, unsigned angle)  
*Get information about a title.*
- void `bd_free_title_info` (`BLURAY_TITLE_INFO *title_info`)  
*Free `BLURAY_TITLE_INFO` object.*
- int `bd_select_title` (`BLURAY *bd`, uint32\_t title)  
*Select the title from the list created by `bd_get_titles()`*
- int `bd_select_playlist` (`BLURAY *bd`, uint32\_t playlist)

- Select a playlist.*

  - `uint32_t bd_get_current_title (BLURAY *bd)`

*Returns the current title index.*
- `int bd_read (BLURAY *bd, unsigned char *buf, int len)`

*Read from currently selected title file, decrypt if possible.*
- `int64_t bd_seek (BLURAY *bd, uint64_t pos)`

*Seek to pos in currently selected title.*
- `int64_t bd_seek_time (BLURAY *bd, uint64_t tick)`

*Seek to specific time in 90Khz ticks.*
- `int64_t bd_seek_chapter (BLURAY *bd, unsigned chapter)`

*Seek to a chapter.*
- `int64_t bd_seek_mark (BLURAY *bd, unsigned mark)`

*Seek to a playmark.*
- `int64_t bd_seek_playitem (BLURAY *bd, unsigned clip_ref)`

*Seek to a playitem.*
- `int bd_select_angle (BLURAY *bd, unsigned angle)`

*Set the angle to play.*
- `void bd_seamless_angle_change (BLURAY *bd, unsigned angle)`

*Initiate seamless angle change.*
- `void bd_select_stream (BLURAY *bd, uint32_t stream_type, uint32_t stream_id, uint32_t enable_flag)`

*Select stream (PG / TextST track)*
- `int64_t bd_chapter_pos (BLURAY *bd, unsigned chapter)`

*Find the byte position of a chapter.*
- `uint32_t bd_get_current_chapter (BLURAY *bd)`

*Get the current chapter.*
- `uint64_t bd_get_title_size (BLURAY *bd)`

*Returns file size in bytes of currently selected title, 0 in no title selected.*
- `unsigned bd_get_current_angle (BLURAY *bd)`

*Return the current angle.*
- `uint64_t bd_tell (BLURAY *bd)`

*Return current pos.*
- `uint64_t bd_tell_time (BLURAY *bd)`

*Return current time.*
- `int bd_set_player_setting (BLURAY *bd, uint32_t idx, uint32_t value)`

*Update player setting.*
- `int bd_set_player_setting_str (BLURAY *bd, uint32_t idx, const char *value)`

*Update player setting (string)*
- `int bd_get_event (BLURAY *bd, BD_EVENT *event)`

*Get event from libbluray event queue.*
- `const char * bd_event_name (uint32_t event)`

*Get name of event.*
- `void bd_register_overlay_proc (BLURAY *bd, void *handle, bd_overlay_proc_f func)`

*Register handler for compressed YUV overlays.*
- `void bd_register_argb_overlay_proc (BLURAY *bd, void *handle, bd_argb_overlay_proc_f func, struct bd_argb↵  
_buffer_s *buf)`

*Register handler for ARGB overlays.*
- `int bd_play (BLURAY *bd)`

- *Start playing disc with on-disc menus.*
- int **bd\_play\_title** (BLURAY \*bd, unsigned title)
- *Play a title (from disc index).*
- int **bd\_menu\_call** (BLURAY \*bd, int64\_t pts)
- *Open BluRay disc Top Menu.*
- int **bd\_read\_ext** (BLURAY \*bd, unsigned char \*buf, int len, BD\_EVENT \*event)
- *Read from currently playing title.*
- int **bd\_read\_skip\_still** (BLURAY \*bd)
- *Continue reading after still mode clip.*
- BLURAY\_TITLE\_INFO \* **bd\_get\_playlist\_info** (BLURAY \*bd, uint32\_t playlist, unsigned angle)
- *Get information about a playlist.*
- int **bd\_get\_sound\_effect** (BLURAY \*bd, unsigned sound\_id, struct bd\_sound\_effect \*effect)
- *Get sound effect.*
- void **bd\_set\_scr** (BLURAY \*bd, int64\_t pts)
- *Update current pts.*
- int **bd\_set\_rate** (BLURAY \*bd, uint32\_t rate)
- *Set current playback rate.*
- int **bd\_user\_input** (BLURAY \*bd, int64\_t pts, uint32\_t key)
- *Pass user input to graphics controller or BD-J.*
- int **bd\_mouse\_select** (BLURAY \*bd, int64\_t pts, uint16\_t x, uint16\_t y)
- *Select menu button at location (x,y).*
- struct clpi\_cl \* **bd\_get\_clpi** (BLURAY \*bd, unsigned clip\_ref)
- *Get copy of clip information for requested playitem.*
- struct clpi\_cl \* **bd\_read\_clpi** (const char \*clpi\_file)
- *Testing/debugging: Parse clip information (CLPI) file.*
- void **bd\_free\_clpi** (struct clpi\_cl \*cl)
- *Free CLPI\_CL object.*
- struct mpls\_pl \* **bd\_read\_mpls** (const char \*mpls\_file)
- *Testing/debugging: Parse playlist (MPLS) file.*
- void **bd\_free\_mpls** (struct mpls\_pl \*)
- *Testing/debugging: Free parsed playlist.*
- struct mobj\_objects \* **bd\_read\_mobj** (const char \*mobj\_file)
- *Testing/debugging: Parse movie objects (MOBJ) file.*
- void **bd\_free\_mobj** (struct mobj\_objects \*)
- *Testing/debugging: Free parsed movie objects.*
- struct bdjo\_data \* **bd\_read\_bdjo** (const char \*bdjo\_file)
- *Testing/debugging: Parse BD-J object file (BDJO)*
- void **bd\_free\_bdjo** (struct bdjo\_data \*)
- *Testing/debugging: Free parsed BDJO object.*
- int **bd\_start\_bdj** (BLURAY \*bd, const char \*start\_object)
- *Testing/debugging: start BD-J from the specified BD-J object (should be a 5 character string)*
- void **bd\_stop\_bdj** (BLURAY \*bd)
- *Testing/debugging: shutdown BD-J and clean up resources.*
- int **bd\_read\_file** (BLURAY \*bd, const char \*path, void \*\*data, int64\_t \*size)
- *Read a file from BluRay Virtual File System.*
- struct bd\_dir\_s \* **bd\_open\_dir** (BLURAY \*bd, const char \*dir)
- *Open a directory from BluRay Virtual File System.*
- struct bd\_file\_s \* **bd\_open\_file\_dec** (BLURAY \*bd, const char \*path)
- *Open a file from BluRay Virtual File System.*

### 5.7.1 Detailed Description

libbluray API

### 5.7.2 Typedef Documentation

#### 5.7.2.1 `bd_argb_overlay_proc_f`

```
typedef void(* bd_argb_overlay_proc_f) (void *handle, const struct bd_argb_overlay_s *const event)
```

ARGB overlay handler function type.

##### Parameters

<i>handle</i>	opaque handle that was given to <a href="#">bd_register_argb_overlay_proc()</a>
<i>event</i>	<a href="#">BD_ARGB_OVERLAY</a> event

#### 5.7.2.2 `bd_overlay_proc_f`

```
typedef void(* bd_overlay_proc_f) (void *handle, const struct bd_overlay_s *const event)
```

YUV overlay handler function type.

##### Parameters

<i>handle</i>	opaque handle that was given to <a href="#">bd_register_overlay_proc()</a>
<i>event</i>	<a href="#">BD_OVERLAY</a> event

#### 5.7.2.3 `BLURAY`

```
typedef struct bluray BLURAY
```

This structure is opaque.

It represents a libbluray instance.

### 5.7.3 Enumeration Type Documentation

#### 5.7.3.1 bd\_event\_e

enum [bd\\_event\\_e](#)

Event type.

Enumerator

BD_EVENT_NONE	no pending events
BD_EVENT_ERROR	Fatal error. Playback can't be continued.
BD_EVENT_READ_ERROR	Reading of .m2ts aligned unit failed. Next call to read will try next block.
BD_EVENT_ENCRYPTED	.m2ts file is encrypted and can't be played
BD_EVENT_ANGLE	current angle, 1...N
BD_EVENT_TITLE	current title, 1...N (0 = top menu)
BD_EVENT_PLAYLIST	current playlist (xxxxx.mpls)
BD_EVENT_PLAYITEM	current play item, 0...N-1
BD_EVENT_CHAPTER	current chapter, 1...N
BD_EVENT_PLAYMARK	playmark reached
BD_EVENT_END_OF_TITLE	end of title reached
BD_EVENT_AUDIO_STREAM	1..32, 0xff = none
BD_EVENT_IG_STREAM	1..32
BD_EVENT_PG_TEXTST_STREAM	1..255, 0xffff = none
BD_EVENT_PIP_PG_TEXTST_STREAM	1..255, 0xffff = none
BD_EVENT_SECONDARY_AUDIO_STREAM	1..32, 0xff = none
BD_EVENT_SECONDARY_VIDEO_STREAM	1..32, 0xff = none
BD_EVENT_PG_TEXTST	0 - disable, 1 - enable
BD_EVENT_PIP_PG_TEXTST	0 - disable, 1 - enable
BD_EVENT_SECONDARY_AUDIO	0 - disable, 1 - enable
BD_EVENT_SECONDARY_VIDEO	0 - disable, 1 - enable
BD_EVENT_SECONDARY_VIDEO_SIZE	0 - PIP, 0xf - fullscreen
BD_EVENT_PLAYLIST_STOP	HDMV VM or JVM stopped playlist playback. Flush all buffers.
BD_EVENT_DISCONTINUITY	discontinuity in the stream (non-seamless connection). Reset demuxer PES buffers. new timestamp (45 kHz)
BD_EVENT_SEEK	HDMV VM or JVM seeked the stream. Next read() will return data from new position. Flush all buffers. new media time (45 kHz)
BD_EVENT_STILL	still playback (pause) 0 - off, 1 - on
BD_EVENT_STILL_TIME	Still playback for n seconds (reached end of still mode play item). Playback continues by calling <a href="#">bd_read_skip_still()</a> . 0 = infinite ; 1...300 = seconds
BD_EVENT_SOUND_EFFECT	Play sound effect. effect ID

## Enumerator

BD_EVENT_IDLE	Nothing to do. Playlist is not playing, but title applet is running. Application should not call bd_read*() immediately again to avoid busy loop.
BD_EVENT_POPUP	Pop-Up menu available. 0 - no, 1 - yes
BD_EVENT_MENU	Interactive menu visible. 0 - no, 1 - yes
BD_EVENT_STEREOSCOPIC_STATUS	3D 0 - 2D, 1 - 3D
BD_EVENT_KEY_INTEREST_TABLE	BD-J key interest table changed. bitmask, BLURAY_KIT_*
BD_EVENT_UO_MASK_CHANGED	UO mask changed. bitmask, BLURAY_UO_*

## 5.7.3.2 bd\_mark\_type\_e

```
enum bd_mark_type_e
```

Mark type.

## Enumerator

BLURAY_MARK_ENTRY	entry mark for chapter search
BLURAY_MARK_LINK	link point

## 5.7.3.3 bd\_player\_setting

```
enum bd_player_setting
```

Player setting.

## Enumerator

BLURAY_PLAYER_SETTING_AUDIO_LANG	Initial audio language. String (ISO 639-2/T).
BLURAY_PLAYER_SETTING_PG_LANG	Initial PG/SPU language. String (ISO 639-2/T).
BLURAY_PLAYER_SETTING_MENU_LANG	Initial menu language. String (ISO 639-2/T).
BLURAY_PLAYER_SETTING_COUNTRY_CODE	Player country code. String (ISO 3166-1/alpha-2).
BLURAY_PLAYER_SETTING_REGION_CODE	Player region code. Integer.
BLURAY_PLAYER_SETTING_OUTPUT_PREFER	Output mode preference. Integer.
BLURAY_PLAYER_SETTING_PARENTAL	Age for parental control. Integer.
BLURAY_PLAYER_SETTING_AUDIO_CAP	Audio capability. Bit mask.
BLURAY_PLAYER_SETTING_VIDEO_CAP	Video capability. Bit mask.
BLURAY_PLAYER_SETTING_DISPLAY_CAP	Display capability. Bit mask.

## Enumerator

BLURAY_PLAYER_SETTING_3D_CAP	3D capability. Bit mask.
BLURAY_PLAYER_SETTING_UHD_CAP	UHD capability.
BLURAY_PLAYER_SETTING_UHD_DISPLAY_CAP	UHD display capability.
BLURAY_PLAYER_SETTING_HDR_PREFERENCE	HDR preference.
BLURAY_PLAYER_SETTING_SDR_CONV_PREFER	SDR conversion preference.
BLURAY_PLAYER_SETTING_TEXT_CAP	Text Subtitle capability. Bit mask.
BLURAY_PLAYER_SETTING_PLAYER_PROFILE	Player profile and version.
BLURAY_PLAYER_SETTING_DECODE_PG	Enable/disable PG (subtitle) decoder. Integer. Default: disabled.
BLURAY_PLAYER_SETTING_PERSISTENT_↔ STORAGE	Enable/disable BD-J persistent storage. Integer. Default: enabled.
BLURAY_PLAYER_PERSISTENT_ROOT	Root path to the BD_J persistent storage location. String.
BLURAY_PLAYER_CACHE_ROOT	Root path to the BD_J cache storage location. String.
BLURAY_PLAYER_JAVA_HOME	Location of JRE. String. Default: NULL (autodetect).

## 5.7.3.4 bd\_still\_mode\_e

```
enum bd_still_mode_e
```

Clip still mode type.

## Enumerator

BLURAY_STILL_NONE	No still (normal playback)
BLURAY_STILL_TIME	Still playback for fixed time.
BLURAY_STILL_INFINITE	Infinite still.

## 5.7.3.5 bd\_video\_format\_e

```
enum bd_video_format_e
```

Stream video format.

## Enumerator

BLURAY_VIDEO_FORMAT_480I	ITU-R BT.601-5.
BLURAY_VIDEO_FORMAT_576I	ITU-R BT.601-4.
BLURAY_VIDEO_FORMAT_480P	SMPTE 293M.
BLURAY_VIDEO_FORMAT_1080I	SMPTE 274M.
BLURAY_VIDEO_FORMAT_720P	SMPTE 296M.
BLURAY_VIDEO_FORMAT_1080P	SMPTE 274M.
BLURAY_VIDEO_FORMAT_576P	ITU-R BT.1358.
BLURAY_VIDEO_FORMAT_2160P	BT.2020.

## 5.7.3.6 bd\_video\_rate\_e

```
enum bd_video_rate_e
```

Stream video frame rate.

## Enumerator

BLURAY_VIDEO_RATE_24000_1001	23.976 Hz
BLURAY_VIDEO_RATE_24	24 Hz
BLURAY_VIDEO_RATE_25	25 Hz
BLURAY_VIDEO_RATE_30000_1001	29.97 Hz
BLURAY_VIDEO_RATE_50	50 Hz
BLURAY_VIDEO_RATE_60000_1001	59.94 Hz

## 5.7.4 Function Documentation

## 5.7.4.1 bd\_chapter\_pos()

```
int64_t bd_chapter_pos (
    BLURAY * bd,
    unsigned chapter )
```

Find the byte position of a chapter.

## Parameters

<i>bd</i>	BLURAY object
<i>chapter</i>	chapter to find position of



**Returns**

seek position of chapter start

**5.7.4.2 bd\_close()**

```
void bd_close (
    BLURAY * bd )
```

Close BluRay disc.

**Parameters**

<i>bd</i>	BLURAY object
-----------	---------------

**5.7.4.3 bd\_event\_name()**

```
const char * bd_event_name (
    uint32_t event )
```

Get name of event.

**Parameters**

<i>event</i>	event type ( <a href="#">bd_event_e</a> )
--------------	---

**Returns**

1 on success, 0 if no events

**5.7.4.4 bd\_free\_clpi()**

```
void bd_free_clpi (
    struct clpi_cl * cl )
```

Free CLPI\_CL object.

## Parameters

<i>cl</i>	CLPI_CL objects
-----------	-----------------

#### 5.7.4.5 `bd_free_title_info()`

```
void bd_free_title_info (
    BLURAY_TITLE_INFO * title_info )
```

Free `BLURAY_TITLE_INFO` object.

## Parameters

<i>title_info</i>	<code>BLURAY_TITLE_INFO</code> object
-------------------	---------------------------------------

#### 5.7.4.6 `bd_get_clpi()`

```
struct clpi_cl * bd_get_clpi (
    BLURAY * bd,
    unsigned clip_ref )
```

Get copy of clip information for requested playitem.

## Parameters

<i>bd</i>	BLURAY objects
<i>clip_ref</i>	requested playitem number

## Returns

pointer to allocated CLPI\_CL object on success, NULL on error

#### 5.7.4.7 `bd_get_current_angle()`

```
unsigned bd_get_current_angle (
    BLURAY * bd )
```

Return the current angle.

**Parameters**

<i>bd</i>	BLURAY object
-----------	---------------

**Returns**

current angle

**5.7.4.8 bd\_get\_current\_chapter()**

```
uint32_t bd_get_current_chapter (
    BLURAY * bd )
```

Get the current chapter.

**Parameters**

<i>bd</i>	BLURAY object
-----------	---------------

**Returns**

current chapter

**5.7.4.9 bd\_get\_current\_title()**

```
uint32_t bd_get_current_title (
    BLURAY * bd )
```

Returns the current title index.

**Parameters**

<i>bd</i>	BLURAY object
-----------	---------------

**Returns**

current title index

#### 5.7.4.10 `bd_get_disc_info()`

```
const BLURAY_DISC_INFO * bd_get_disc_info (
    BLURAY * bd )
```

Get information about current BluRay disc.

##### Parameters

<i>bd</i>	BLURAY object
-----------	---------------

##### Returns

pointer to [BLURAY\\_DISC\\_INFO](#) object, NULL on error

#### 5.7.4.11 `bd_get_event()`

```
int bd_get_event (
    BLURAY * bd,
    BD_EVENT * event )
```

Get event from libbluray event queue.

##### Parameters

<i>bd</i>	BLURAY object
<i>event</i>	next <a href="#">BD_EVENT</a> from event queue, NULL to initialize event queue

##### Returns

1 on success, 0 if no events

#### 5.7.4.12 `bd_get_main_title()`

```
int bd_get_main_title (
    BLURAY * bd )
```

Get main title Returned number is an index to the list created by [bd\\_get\\_titles\(\)](#)

## Parameters

<i>bd</i>	BLURAY object
-----------	---------------

## Returns

title index of main title, -1 on error

#### 5.7.4.13 `bd_get_meta()`

```
const struct meta_dl * bd_get_meta (
    BLURAY * bd )
```

Get meta information about current BluRay disc.

Meta information is optional in BluRay discs. If information is provided in multiple languages, currently selected language (BLURAY\_PLAYER\_SETTING\_MENU\_LANG) is used.

Referenced thumbnail images should be read with [bd\\_get\\_meta\\_file\(\)](#).

## Parameters

<i>bd</i>	BLURAY object
-----------	---------------

## Returns

[META\\_DL](#) (disclib) object, NULL on error

#### 5.7.4.14 `bd_get_meta_file()`

```
int bd_get_meta_file (
    BLURAY * bd,
    const char * file_name,
    void ** data,
    int64_t * size )
```

Read metadata file from BluRay disc.

Allocate large enough memory block and read file contents. Caller must free the memory block with `free()`.

## Parameters

<i>bd</i>	BLURAY object
<i>file_name</i>	name of metadata file
<i>data</i>	where to store pointer to file data
<i>size</i>	where to store file size

## Returns

1 on success, 0 on error

**5.7.4.15 bd\_get\_playlist\_info()**

```
BLURAY_TITLE_INFO * bd_get_playlist_info (
    BLURAY * bd,
    uint32_t playlist,
    unsigned angle )
```

Get information about a playlist.

## Parameters

<i>bd</i>	BLURAY object
<i>playlist</i>	playlist number
<i>angle</i>	angle number (chapter offsets and clip size depend on selected angle)

## Returns

allocated [BLURAY\\_TITLE\\_INFO](#) object, NULL on error

**5.7.4.16 bd\_get\_sound\_effect()**

```
int bd_get_sound_effect (
    BLURAY * bd,
    unsigned sound_id,
    struct bd_sound_effect * effect )
```

Get sound effect.

## Parameters

<i>bd</i>	BLURAY object
<i>sound↔ _id</i>	sound effect id (0...N)
<i>effect</i>	sound effect data

## Returns

<0 when no effects, 0 when id out of range, 1 on success

5.7.4.17 **bd\_get\_title\_info()**

```
BLURAY_TITLE_INFO * bd_get_title_info (
    BLURAY * bd,
    uint32_t title_idx,
    unsigned angle )
```

Get information about a title.

## Parameters

<i>bd</i>	BLURAY object
<i>title_idx</i>	title index number
<i>angle</i>	angle number (chapter offsets and clip size depend on selected angle)

## Returns

allocated [BLURAY\\_TITLE\\_INFO](#) object, NULL on error

5.7.4.18 **bd\_get\_title\_size()**

```
uint64_t bd_get_title_size (
    BLURAY * bd )
```

Returns file size in bytes of currently selected title, 0 in no title selected.

## Parameters

<i>bd</i>	BLURAY object
-----------	---------------

**Returns**

file size in bytes of currently selected title, 0 if no title selected

**5.7.4.19 bd\_get\_titles()**

```
uint32_t bd_get_titles (
    BLURAY * bd,
    uint8_t flags,
    uint32_t min_title_length )
```

Get number of titles (playlists)

This must be called after [bd\\_open\(\)](#) and before [bd\\_select\\_title\(\)](#). Populates the title list in BLURAY. Filtering of the returned list is controled through title flags

**Parameters**

<i>bd</i>	BLURAY object
<i>flags</i>	title flags
<i>min_title_length</i>	filter out titles shorter than min_title_length seconds

**Returns**

number of titles found

**5.7.4.20 bd\_get\_version()**

```
void bd_get_version (
    int * major,
    int * minor,
    int * micro )
```

Get libbluray version.

Get the version of libbluray (runtime)

See also [bluray-version.h](#)

**Parameters**

<i>major</i>	where to store major version
<i>minor</i>	where to store minor version
<i>micro</i>	where to store micro version



#### 5.7.4.21 bd\_init()

```
BLURAY * bd_init (
    void )
```

Initialize BLURAY object.

Resulting object can be passed to following bd\_open\_??? functions.

##### Returns

allocated BLURAY object, NULL if error

#### 5.7.4.22 bd\_menu\_call()

```
int bd_menu_call (
    BLURAY * bd,
    int64_t pts )
```

Open BluRay disc Top Menu.

Current pts is needed for resuming playback when menu is closed.

##### Parameters

<i>bd</i>	BLURAY object
<i>pts</i>	current playback position (1/90000s) or -1

##### Returns

1 on success, 0 if error

#### 5.7.4.23 bd\_mouse\_select()

```
int bd_mouse_select (
    BLURAY * bd,
    int64_t pts,
    uint16_t x,
    uint16_t y )
```

Select menu button at location (x,y).

This function has no effect with BD-J menus.

**Parameters**

<i>bd</i>	BLURAY object
<i>pts</i>	current playback position (1/90000s) or -1
<i>x</i>	mouse pointer x-position
<i>y</i>	mouse pointer y-position

**Returns**

<0 on error, 0 when mouse is outside of buttons, 1 when mouse is inside button

**5.7.4.24 bd\_open()**

```
BLURAY * bd_open (
    const char * device_path,
    const char * keyfile_path )
```

Open BluRay disc.

Shortcut for bd\_open\_disc(bd\_init(), device\_path, keyfile\_path)

**Parameters**

<i>device_path</i>	path to mounted Blu-ray disc, device or image file
<i>keyfile_path</i>	path to KEYDB.cfg (may be NULL)

**Returns**

allocated BLURAY object, NULL if error

**5.7.4.25 bd\_open\_dir()**

```
struct bd_dir_s * bd_open_dir (
    BLURAY * bd,
    const char * dir )
```

Open a directory from BluRay Virtual File System.

Caller must close with dir->close().

## Parameters

<i>bd</i>	BLURAY object
<i>dir</i>	target directory (relative to disc root)

## Returns

BD\_DIR\_H \*, NULL if failed

**5.7.4.26 bd\_open\_disc()**

```
int bd_open_disc (
    BLURAY * bd,
    const char * device_path,
    const char * keyfile_path )
```

Open BluRay disc.

## Parameters

<i>bd</i>	BLURAY object
<i>device_path</i>	path to mounted Blu-ray disc, device or image file
<i>keyfile_path</i>	path to KEYDB.cfg (may be NULL)

## Returns

1 on success, 0 if error

**5.7.4.27 bd\_open\_file\_dec()**

```
struct bd_file_s * bd_open_file_dec (
    BLURAY * bd,
    const char * path )
```

Open a file from BluRay Virtual File System.

encrypted streams are decrypted, and because of how decryption works, it can only seek to (N\*6144) bytes, and read 6144 bytes at a time. DO NOT mix any play functionalities with these functions. It might cause broken stream. In general, accessing multiple file on disk at the same time is a bad idea.

Caller must close with file->close().

**Parameters**

<i>bd</i>	BLURAY object
<i>path</i>	path to the file (relative to disc root)

**Returns**

BD\_FILE\_H \*, NULL if failed

**5.7.4.28 bd\_open\_files()**

```
int bd_open_files (
    BLURAY * bd,
    void * handle,
    struct bd_dir_s *(*)(void *handle, const char *rel_path) open_dir,
    struct bd_file_s *(*)(void *handle, const char *rel_path) open_file )
```

Open BluRay disc.

**Parameters**

<i>bd</i>	BLURAY object
<i>handle</i>	opaque handle for open_dir and open_file
<i>open_dir</i>	function used to open a directory
<i>open_file</i>	function used to open a file

**Returns**

1 on success, 0 if error

**5.7.4.29 bd\_open\_stream()**

```
int bd_open_stream (
    BLURAY * bd,
    void * read_blocks_handle,
    int(*)(void *handle, void *buf, int lba, int num_blocks) read_blocks )
```

Open BluRay disc.

## Parameters

<i>bd</i>	BLURAY object
<i>read_blocks_handle</i>	opaque handle for read_blocks
<i>read_blocks</i>	function used to read disc blocks

## Returns

1 on success, 0 if error

**5.7.4.30 bd\_play()**

```
int bd_play (
    BLURAY * bd )
```

Start playing disc with on-disc menus.

Playback is started from "First Play" title.

## Parameters

<i>bd</i>	BLURAY object
-----------	---------------

## Returns

1 on success, 0 if error

**5.7.4.31 bd\_play\_title()**

```
int bd_play_title (
    BLURAY * bd,
    unsigned title )
```

Play a title (from disc index).

Title 0 = Top Menu Title 0xffff = First Play title Number of titles can be found from [BLURAY\\_DISC\\_INFO](#).

## Parameters

<i>bd</i>	BLURAY object
<i>title</i>	title number from disc index

**Returns**

1 on success, 0 if error

**5.7.4.32 bd\_read()**

```
int bd_read (
    BLURAY * bd,
    unsigned char * buf,
    int len )
```

Read from currently selected title file, decrypt if possible.

**Parameters**

<i>bd</i>	BLURAY object
<i>buf</i>	buffer to read data into
<i>len</i>	size of data to be read

**Returns**

size of data read, -1 if error, 0 if EOF

**5.7.4.33 bd\_read\_ext()**

```
int bd_read_ext (
    BLURAY * bd,
    unsigned char * buf,
    int len,
    BD_EVENT * event )
```

Read from currently playing title.

When playing disc in navigation mode this function must be used instead of [bd\\_read\(\)](#).

**Parameters**

<i>bd</i>	BLURAY object
<i>buf</i>	buffer to read data into
<i>len</i>	size of data to be read
<i>event</i>	next <a href="#">BD_EVENT</a> from event queue (BD_EVENT_NONE if no events)

**Returns**

size of data read, -1 if error, 0 if event needs to be handled first, 0 if end of title was reached

**5.7.4.34 bd\_read\_file()**

```
int bd_read_file (
    BLURAY * bd,
    const char * path,
    void ** data,
    int64_t * size )
```

Read a file from BluRay Virtual File System.

Allocate large enough memory block and read file contents. Caller must free the memory block with free().

**Parameters**

<i>bd</i>	BLURAY object
<i>path</i>	path to the file (relative to disc root)
<i>data</i>	where to store pointer to allocated data
<i>size</i>	where to store file size

**Returns**

1 on success, 0 on error

**5.7.4.35 bd\_read\_skip\_still()**

```
int bd_read_skip_still (
    BLURAY * bd )
```

Continue reading after still mode clip.

**Parameters**

<i>bd</i>	BLURAY object
-----------	---------------

**Returns**

0 on error

#### 5.7.4.36 bd\_register\_argb\_overlay\_proc()

```
void bd_register_argb_overlay_proc (
    BLURAY * bd,
    void * handle,
    bd_argb_overlay_proc_f func,
    struct bd_argb_buffer_s * buf )
```

Register handler for ARGB overlays.

ARGB overlays are used with BD-J (Java) menus.

Callback function can be called at any time by a thread created by Java VM. No more than single call for each overlay plane are executed in parallel.

##### Parameters

<i>bd</i>	BLURAY object
<i>handle</i>	application-specific handle that will be passed to handler function
<i>func</i>	handler function pointer
<i>buf</i>	optional application-allocated frame buffer

##### Returns

1 on success, 0 if error

#### 5.7.4.37 bd\_register\_overlay\_proc()

```
void bd_register_overlay_proc (
    BLURAY * bd,
    void * handle,
    bd_overlay_proc_f func )
```

Register handler for compressed YUV overlays.

Compressed YUV overlays are used with presentation graphics (subtitles) and HDMV mode menus. This function can be used when player does not support full-screen ARGB overlays or player can optimize drawing of compressed overlays, color space conversion etc.

Callback function is called from application thread context while `bd_*` functions are called.

Note that BD-J mode outputs only ARGB graphics.

##### Parameters

<i>bd</i>	BLURAY object
<i>handle</i>	application-specific handle that will be passed to handler function
<i>func</i>	handler function pointer



**Returns**

1 on success, 0 if error

**5.7.4.38 bd\_seamless\_angle\_change()**

```
void bd_seamless_angle_change (
    BLURAY * bd,
    unsigned angle )
```

Initiate seamless angle change.

**Parameters**

<i>bd</i>	BLURAY object
<i>angle</i>	angle to change to

**5.7.4.39 bd\_seek()**

```
int64_t bd_seek (
    BLURAY * bd,
    uint64_t pos )
```

Seek to pos in currently selected title.

**Parameters**

<i>bd</i>	BLURAY object
<i>pos</i>	position to seek to

**Returns**

current seek position

**5.7.4.40 bd\_seek\_chapter()**

```
int64_t bd_seek_chapter (
    BLURAY * bd,
    unsigned chapter )
```

Seek to a chapter.

First chapter is 0

## Parameters

<i>bd</i>	BLURAY object
<i>chapter</i>	chapter to seek to

## Returns

current seek position

**5.7.4.41 bd\_seek\_mark()**

```
int64_t bd_seek_mark (
    BLURAY * bd,
    unsigned mark )
```

Seek to a playmark.

First mark is 0

## Parameters

<i>bd</i>	BLURAY object
<i>mark</i>	playmark to seek to

## Returns

current seek position

**5.7.4.42 bd\_seek\_playitem()**

```
int64_t bd_seek_playitem (
    BLURAY * bd,
    unsigned clip_ref )
```

Seek to a playitem.

## Parameters

<i>bd</i>	BLURAY object
<i>clip_ref</i>	playitem to seek to

**Returns**

current seek position

**5.7.4.43 bd\_seek\_time()**

```
int64_t bd_seek_time (
    BLURAY * bd,
    uint64_t tick )
```

Seek to specific time in 90Khz ticks.

**Parameters**

<i>bd</i>	BLURAY object
<i>tick</i>	tick count

**Returns**

current seek position

**5.7.4.44 bd\_select\_angle()**

```
int bd_select_angle (
    BLURAY * bd,
    unsigned angle )
```

Set the angle to play.

**Parameters**

<i>bd</i>	BLURAY object
<i>angle</i>	angle to play

**Returns**

1 on success, 0 if error

#### 5.7.4.45 bd\_select\_playlist()

```
int bd_select_playlist (
    BLURAY * bd,
    uint32_t playlist )
```

Select a playlist.

##### Parameters

<i>bd</i>	BLURAY object
<i>playlist</i>	playlist to select

##### Returns

1 on success, 0 if error

#### 5.7.4.46 bd\_select\_stream()

```
void bd_select_stream (
    BLURAY * bd,
    uint32_t stream_type,
    uint32_t stream_id,
    uint32_t enable_flag )
```

Select stream (PG / TextST track)

When playing with on-disc menus:

Stream selection is controlled by on-disc menus. If user can change stream selection also in player GUI, this function should be used to keep on-disc menus in sync with player GUI.

When playing the disc without on-disc menus:

Initial stream selection is done using preferred language settings. This function can be used to override automatic stream selection. Without on-disc menus selecting the stream is useful only when using libbluray internal decoders or the stream is stored in a sub-path.

##### Parameters

<i>bd</i>	BLURAY object
<i>stream_type</i>	BLURAY_AUDIO_STREAM or BLURAY_PG_TEXTST_STREAM
<i>stream_id</i>	stream number (1..N)
<i>enable_flag</i>	set to 0 to disable streams of this type

#### 5.7.4.47 `bd_select_title()`

```
int bd_select_title (
    BLURAY * bd,
    uint32_t title )
```

Select the title from the list created by [bd\\_get\\_titles\(\)](#)

##### Parameters

<i>bd</i>	BLURAY object
<i>title</i>	title to select

##### Returns

1 on success, 0 if error

#### 5.7.4.48 `bd_set_player_setting()`

```
int bd_set_player_setting (
    BLURAY * bd,
    uint32_t idx,
    uint32_t value )
```

Update player setting.

Bit masks and enumeration values are defined in [player\\_settings.h](#).

##### Parameters

<i>bd</i>	BLURAY object
<i>idx</i>	Player setting to update
<i>value</i>	New value for player setting

##### Returns

1 on success, 0 on error (invalid setting)

#### 5.7.4.49 bd\_set\_player\_setting\_str()

```
int bd_set_player_setting_str (
    BLURAY * bd,
    uint32_t idx,
    const char * value )
```

Update player setting (string)

##### Parameters

<i>bd</i>	BLURAY object
<i>idx</i>	Player setting to update
<i>value</i>	New value for player setting

##### Returns

1 on success, 0 on error (invalid setting)

#### 5.7.4.50 bd\_set\_rate()

```
int bd_set_rate (
    BLURAY * bd,
    uint32_t rate )
```

Set current playback rate.

Notify BD-J media player when user changes playback rate (ex. pauses playback). Changing rate may fail if corresponding UO is masked or playlist is not playing.

##### Parameters

<i>bd</i>	BLURAY object
<i>rate</i>	current playback rate * 90000 (0 = paused, 90000 = normal)

##### Returns

<0 on error, 0 on success

#### 5.7.4.51 bd\_set\_scr()

```
void bd_set_scr (
    BLURAY * bd,
    int64_t pts )
```

Update current pts.

#### Parameters

<i>bd</i>	BLURAY object
<i>pts</i>	current playback position (1/90000s) or -1

#### 5.7.4.52 `bd_tell()`

```
uint64_t bd_tell (
    BLURAY * bd )
```

Return current pos.

#### Parameters

<i>bd</i>	BLURAY object
-----------	---------------

#### Returns

current seek position

#### 5.7.4.53 `bd_tell_time()`

```
uint64_t bd_tell_time (
    BLURAY * bd )
```

Return current time.

#### Parameters

<i>bd</i>	BLURAY object
-----------	---------------

#### Returns

current time



#### 5.7.4.54 bd\_user\_input()

```
int bd_user_input (
    BLURAY * bd,
    int64_t pts,
    uint32_t key )
```

Pass user input to graphics controller or BD-J.

Keys are defined in [libbluray/keys.h](#).

Two user input models are supported:

- Single event when a key is typed once.
- Separate events when key is pressed and released. VD\_VK\_KEY\_PRESSED, BD\_VK\_TYPED and BD\_VK\_↵ KEY\_RELEASED are or'd with the key.

##### Parameters

<i>bd</i>	BLURAY object
<i>pts</i>	current playback position (1/90000s) or -1
<i>key</i>	input key (

##### See also

[keys.h](#))

##### Returns

<0 on error, 0 on success, >0 if selection/activation changed

## 5.8 bluray.h

[Go to the documentation of this file.](#)

```
1 /*
2  * This file is part of libbluray
3  * Copyright (C) 2009-2010 Obliter0n
4  * Copyright (C) 2009-2010 John Stebbins
5  * Copyright (C) 2010-2017 Petri Hintukainen
6  *
7  * This library is free software; you can redistribute it and/or
8  * modify it under the terms of the GNU Lesser General Public
9  * License as published by the Free Software Foundation; either
10 * version 2.1 of the License, or (at your option) any later version.
11 *
12 * This library is distributed in the hope that it will be useful,
13 * but WITHOUT ANY WARRANTY; without even the implied warranty of
14 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
15 * Lesser General Public License for more details.
16 *
17 * You should have received a copy of the GNU Lesser General Public
18 * License along with this library. If not, see
```

```

19  * <http://www.gnu.org/licenses/>.
20  */
21
22 #ifndef BLURAY_H_
23 #define BLURAY_H_
24
25 #ifdef __cplusplus
26 extern "C" {
27 #endif
28
29 #include <stdint.h>
30
31 #define TITLES_ALL 0
32 #define TITLES_FILTER_DUP_TITLE 0x01
33 #define TITLES_FILTER_DUP_CLIP 0x02
34 #define TITLES_RELEVANT \
35 (TITLES_FILTER_DUP_TITLE | TITLES_FILTER_DUP_CLIP)
36 typedef struct bluray BLURAY;
37
38 /*
39  * Disc info
40  */
41
42 /* AACS error codes */
43 #define BD_AACS_CORRUPTED_DISC -1
44 #define BD_AACS_NO_CONFIG -2
45 #define BD_AACS_NO_PK -3
46 #define BD_AACS_NO_CERT -4
47 #define BD_AACS_CERT_REVOKED -5
48 #define BD_AACS_MMC_FAILED -6
49 typedef struct {
50     const char *name;
51     uint8_t interactive;
52     uint8_t accessible;
53     uint8_t hidden;
54     uint8_t bdj;
55     uint32_t id_ref;
56 } BLURAY_TITLE;
57
58 typedef struct {
59     uint8_t bluray_detected;
60     /* Disc ID */
61     const char *disc_name;
62     const char *udf_volume_id;
63     uint8_t disc_id[20];
64     uint8_t no_menu_support;
65     uint8_t first_play_supported;
66     uint8_t top_menu_supported;
67     uint32_t num_titles;
68     const BLURAY_TITLE *const *titles;
69     const BLURAY_TITLE *first_play;
70     const BLURAY_TITLE *top_menu;
71     uint32_t num_hdmv_titles;
72     uint32_t num_bdj_titles;
73     uint32_t num_unsupported_titles;
74     uint8_t bdj_detected;
75     uint8_t bdj_supported;
76     uint8_t libjvm_detected;
77     uint8_t bdj_handled;
78     char bdj_org_id[9];
79     char bdj_disc_id[33];
80     /* disc application info */
81     uint8_t video_format;
82     uint8_t frame_rate;
83     uint8_t content_exist_3D;
84     uint8_t initial_output_mode_preference;
85     uint8_t provider_data[32];
86     /* AACS info (valid only if disc uses AACS) */
87     uint8_t aacs_detected;
88     uint8_t libaacs_detected;
89     uint8_t aacs_handled;
90     int aacs_error_code;
91     int aacs_mkbv;
92     /* BD+ info (valid only if disc uses BD+) */
93     uint8_t bdplus_detected;
94     uint8_t libbdplus_detected;
95     uint8_t bdplus_handled;
96     uint8_t bdplus_gen;
97     uint32_t bdplus_date;
98     /* disc application info (libbluray > 1.2.0) */
99     uint8_t initial_dynamic_range_type;

```

```

128 } BLURAY_DISC_INFO;
129
130 /*
131  * Playlist info
132  */
133
134 typedef enum {
135     BLURAY_STREAM_TYPE_VIDEO_MPEG1          = 0x01,
136     BLURAY_STREAM_TYPE_VIDEO_MPEG2          = 0x02,
137     BLURAY_STREAM_TYPE_AUDIO_MPEG1          = 0x03,
138     BLURAY_STREAM_TYPE_AUDIO_MPEG2          = 0x04,
139     BLURAY_STREAM_TYPE_AUDIO_LPCM           = 0x80,
140     BLURAY_STREAM_TYPE_AUDIO_AC3            = 0x81,
141     BLURAY_STREAM_TYPE_AUDIO_DTS            = 0x82,
142     BLURAY_STREAM_TYPE_AUDIO_TRUHD          = 0x83,
143     BLURAY_STREAM_TYPE_AUDIO_AC3PLUS        = 0x84,
144     BLURAY_STREAM_TYPE_AUDIO_DTSHD          = 0x85,
145     BLURAY_STREAM_TYPE_AUDIO_DTSHD_MASTER   = 0x86,
146     BLURAY_STREAM_TYPE_VIDEO_VC1            = 0xea,
147     BLURAY_STREAM_TYPE_VIDEO_H264           = 0x1b,
148     BLURAY_STREAM_TYPE_VIDEO_HEVC           = 0x24,
149     BLURAY_STREAM_TYPE_SUB_PG                = 0x90,
150     BLURAY_STREAM_TYPE_SUB_IG                = 0x91,
151     BLURAY_STREAM_TYPE_SUB_TEXT              = 0x92,
152     BLURAY_STREAM_TYPE_AUDIO_AC3PLUS_SECONDARY = 0xa1,
153     BLURAY_STREAM_TYPE_AUDIO_DTSHD_SECONDARY = 0xa2
154 } bd_stream_type_e;
155
156 typedef enum {
157     BLURAY_VIDEO_FORMAT_480I                = 1,
158     BLURAY_VIDEO_FORMAT_576I                = 2,
159     BLURAY_VIDEO_FORMAT_480P                = 3,
160     BLURAY_VIDEO_FORMAT_1080I               = 4,
161     BLURAY_VIDEO_FORMAT_720P                = 5,
162     BLURAY_VIDEO_FORMAT_1080P               = 6,
163     BLURAY_VIDEO_FORMAT_576P                = 7,
164     BLURAY_VIDEO_FORMAT_2160P               = 8,
165 } bd_video_format_e;
166
167 typedef enum {
168     BLURAY_VIDEO_RATE_24000_1001           = 1,
169     BLURAY_VIDEO_RATE_24                   = 2,
170     BLURAY_VIDEO_RATE_25                   = 3,
171     BLURAY_VIDEO_RATE_30000_1001           = 4,
172     BLURAY_VIDEO_RATE_50                   = 6,
173     BLURAY_VIDEO_RATE_60000_1001           = 7,
174 } bd_video_rate_e;
175
176 typedef enum {
177     BLURAY_ASPECT_RATIO_4_3                 = 2,
178     BLURAY_ASPECT_RATIO_16_9                = 3,
179 } bd_video_aspect_e;
180
181 typedef enum {
182     BLURAY_AUDIO_FORMAT_MONO                = 1,
183     BLURAY_AUDIO_FORMAT_STEREO              = 3,
184     BLURAY_AUDIO_FORMAT_MULTI_CHAN          = 6,
185     BLURAY_AUDIO_FORMAT_COMBO               = 12 // Stereo ac3/dts,
186 } bd_audio_format_e;
187
188 // multi mlp/dts-hd
189
190 typedef enum {
191     BLURAY_AUDIO_RATE_48                    = 1,
192     BLURAY_AUDIO_RATE_96                    = 4,
193     BLURAY_AUDIO_RATE_192                   = 5,
194     BLURAY_AUDIO_RATE_192_COMBO             = 12, // 48 or 96 ac3/dts
195     // 192 mlp/dts-hd
196     BLURAY_AUDIO_RATE_96_COMBO              = 14 // 48 ac3/dts
197     // 96 mlp/dts-hd
198 } bd_audio_rate_e;
199
200 typedef enum {
201     BLURAY_TEXT_CHAR_CODE_UTF8              = 0x01,
202     BLURAY_TEXT_CHAR_CODE_UTF16BE          = 0x02,
203     BLURAY_TEXT_CHAR_CODE_SHIFT_JIS         = 0x03,
204     BLURAY_TEXT_CHAR_CODE_EUC_KR           = 0x04,
205     BLURAY_TEXT_CHAR_CODE_GB18030_20001    = 0x05,
206     BLURAY_TEXT_CHAR_CODE_CN_GB            = 0x06,
207     BLURAY_TEXT_CHAR_CODE_BIG5              = 0x07,
208 } bd_char_code_e;
209
210 typedef enum {

```

```

217     BLURAY_STILL_NONE      = 0x00,
218     BLURAY_STILL_TIME      = 0x01,
219     BLURAY_STILL_INFINITE  = 0x02,
220 } bd_still_mode_e;
221
222 typedef enum {
223     BLURAY_MARK_ENTRY      = 0x01,
224     BLURAY_MARK_LINK       = 0x02,
225 } bd_mark_type_e;
226
227 typedef enum {
228     BLURAY_DYNAMIC_RANGE_SDR      = 0,
229     BLURAY_DYNAMIC_RANGE_HDR10    = 1,
230     BLURAY_DYNAMIC_RANGE_DOLBY_VISION = 2
231 } bd_dynamic_range_type_e;
232
233 typedef struct bd_stream_info {
234     uint8_t      coding_type;
235     uint8_t      format;
236     uint8_t      rate;
237     uint8_t      char_code;
238     uint8_t      lang[4];
239     uint16_t     pid;
240     uint8_t      aspect;
241     uint8_t      subpath_id;
242 } BLURAY_STREAM_INFO;
243
244 typedef struct bd_clip {
245     uint32_t      pkt_count;
246     uint8_t      still_mode;
247     uint16_t      still_time;
248     uint8_t      video_stream_count;
249     uint8_t      audio_stream_count;
250     uint8_t      pg_stream_count;
251     uint8_t      ig_stream_count;
252     uint8_t      sec_audio_stream_count;
253     uint8_t      sec_video_stream_count;
254     BLURAY_STREAM_INFO *video_streams;
255     BLURAY_STREAM_INFO *audio_streams;
256     BLURAY_STREAM_INFO *pg_streams;
257     BLURAY_STREAM_INFO *ig_streams;
258     BLURAY_STREAM_INFO *sec_audio_streams;
259     BLURAY_STREAM_INFO *sec_video_streams;
260     uint64_t      start_time;
261     uint64_t      in_time;
262     uint64_t      out_time;
263     char          clip_id[6];
264 } BLURAY_CLIP_INFO;
265
266 typedef struct bd_chapter {
267     uint32_t      idx;
268     uint64_t      start;
269     uint64_t      duration;
270     uint64_t      offset;
271     unsigned      clip_ref;
272 } BLURAY_TITLE_CHAPTER;
273
274 typedef struct bd_mark {
275     uint32_t      idx;
276     int           type;
277     uint64_t      start;
278     uint64_t      duration;
279     uint64_t      offset;
280     unsigned      clip_ref;
281 } BLURAY_TITLE_MARK;
282
283 typedef struct bd_title_info {
284     uint32_t      idx;
285     uint32_t      playlist;
286     uint64_t      duration;
287     uint32_t      clip_count;
288     uint8_t      angle_count;
289     uint32_t      chapter_count;
290     uint32_t      mark_count;
291     BLURAY_CLIP_INFO *clips;
292     BLURAY_TITLE_CHAPTER *chapters;
293     BLURAY_TITLE_MARK *marks;
294     uint8_t      mvc_base_view_r_flag;
295 } BLURAY_TITLE_INFO;
296
297 typedef struct bd_sound_effect {

```

```

308     uint8_t         num_channels;
309     uint32_t        num_frames;
310     const int16_t   *samples;
311 } BLURAY_SOUND_EFFECT;
312
313
325 void bd_get_version(int *major, int *minor, int *micro);
326
327 /*
328  * Disc functions
329  */
330
331 struct bd_dir_s;
332 struct bd_file_s;
333 struct meta_dl;
334
335 BLURAY *bd_open(const char *device_path, const char *keyfile_path);
336
337 BLURAY *bd_init(void);
338
339 int bd_open_disc(BLURAY *bd, const char *device_path, const char *keyfile_path);
340
341 int bd_open_stream(BLURAY *bd,
342                   void *read_blocks_handle,
343                   int (*read_blocks)(void *handle, void *buf, int lba, int num_blocks));
344
345 int bd_open_files(BLURAY *bd,
346                  void *handle,
347                  struct bd_dir_s *(*open_dir)(void *handle, const char *rel_path),
348                  struct bd_file_s *(*open_file)(void *handle, const char *rel_path));
349
350 void bd_close(BLURAY *bd);
351
352 const BLURAY_DISC_INFO *bd_get_disc_info(BLURAY *bd);
353
354 const struct meta_dl *bd_get_meta(BLURAY *bd);
355
356 int bd_get_meta_file(BLURAY *bd, const char *file_name, void **data, int64_t *size);
357
358 /*
359  * Title selection without on-disc menus
360  */
361
362 uint32_t bd_get_titles(BLURAY *bd, uint8_t flags, uint32_t min_title_length);
363
364 int bd_get_main_title(BLURAY *bd);
365
366 BLURAY_TITLE_INFO* bd_get_title_info(BLURAY *bd, uint32_t title_idx, unsigned angle);
367
368 void bd_free_title_info(BLURAY_TITLE_INFO *title_info);
369
370 int bd_select_title(BLURAY *bd, uint32_t title);
371
372 int bd_select_playlist(BLURAY *bd, uint32_t playlist);
373
374 uint32_t bd_get_current_title(BLURAY *bd);
375
376 int bd_read(BLURAY *bd, unsigned char *buf, int len);
377
378 /*
379  * Playback control functions
380  */
381
382 int64_t bd_seek(BLURAY *bd, uint64_t pos);
383
384 int64_t bd_seek_time(BLURAY *bd, uint64_t tick);
385
386 int64_t bd_seek_chapter(BLURAY *bd, unsigned chapter);
387
388 int64_t bd_seek_mark(BLURAY *bd, unsigned mark);
389
390 int64_t bd_seek_playitem(BLURAY *bd, unsigned clip_ref);
391
392 int bd_select_angle(BLURAY *bd, unsigned angle);
393
394 void bd_seamless_angle_change(BLURAY *bd, unsigned angle);
395
396 void bd_select_stream(BLURAY *bd, uint32_t stream_type, uint32_t stream_id, uint32_t enable_flag);
397
398

```

```

623 #define BLURAY_AUDIO_STREAM      0
624 #define BLURAY_PG_TEXTST_STREAM  1
627 /*
628  * Playback status functions
629  */
630
631 int64_t bd_chapter_pos(BLURAY *bd, unsigned chapter);
632
633 uint32_t bd_get_current_chapter(BLURAY *bd);
634
635 uint64_t bd_get_title_size(BLURAY *bd);
636
637 unsigned bd_get_current_angle(BLURAY *bd);
638
639 uint64_t bd_tell(BLURAY *bd);
640
641 uint64_t bd_tell_time(BLURAY *bd);
642
643 /*
644  * player settings
645  */
646
647 typedef enum {
648     BLURAY_PLAYER_SETTING_AUDIO_LANG      = 16,
649     BLURAY_PLAYER_SETTING_PG_LANG        = 17,
650     BLURAY_PLAYER_SETTING_MENU_LANG      = 18,
651     BLURAY_PLAYER_SETTING_COUNTRY_CODE   = 19,
652     BLURAY_PLAYER_SETTING_REGION_CODE    = 20,
653     BLURAY_PLAYER_SETTING_OUTPUT_PREFER  = 21,
654     BLURAY_PLAYER_SETTING_PARENTAL       = 13,
655     BLURAY_PLAYER_SETTING_AUDIO_CAP      = 15,
656     BLURAY_PLAYER_SETTING_VIDEO_CAP      = 29,
657     BLURAY_PLAYER_SETTING_DISPLAY_CAP    = 23,
658     BLURAY_PLAYER_SETTING_3D_CAP         = 24,
659     BLURAY_PLAYER_SETTING_UHD_CAP        = 25,
660     BLURAY_PLAYER_SETTING_UHD_DISPLAY_CAP = 26,
661     BLURAY_PLAYER_SETTING_HDR_PREFERENCE = 27,
662     BLURAY_PLAYER_SETTING_SDR_CONV_PREFER = 28,
663     BLURAY_PLAYER_SETTING_TEXT_CAP       = 30,
664     BLURAY_PLAYER_SETTING_PLAYER_PROFILE = 31,
665     BLURAY_PLAYER_SETTING_DECODE_PG      = 0x100,
666     BLURAY_PLAYER_SETTING_PERSISTENT_STORAGE = 0x101,
667     BLURAY_PLAYER_PERSISTENT_ROOT        = 0x200,
668     BLURAY_PLAYER_CACHE_ROOT             = 0x201,
669     BLURAY_PLAYER_JAVA_HOME              = 0x202,
670 } bd_player_setting;
671
672 int bd_set_player_setting(BLURAY *bd, uint32_t idx, uint32_t value);
673
674 int bd_set_player_setting_str(BLURAY *bd, uint32_t idx, const char *value);
675
676 /*
677  * events
678  */
679
680 typedef enum {
681     BD_EVENT_NONE          = 0,
682     /*
683      * errors
684      */
685     BD_EVENT_ERROR         = 1,
686     BD_EVENT_READ_ERROR    = 2,
687     BD_EVENT_ENCRYPTED      = 3,
688     /*
689      * current playback position
690      */
691     BD_EVENT_ANGLE         = 4,
692     BD_EVENT_TITLE         = 5,
693     BD_EVENT_PLAYLIST      = 6,
694     BD_EVENT_PLAYITEM      = 7,
695     BD_EVENT_CHAPTER       = 8,
696     BD_EVENT_PLAYMARK      = 9,
697     BD_EVENT_END_OF_TITLE  = 10,
698     /*
699      * stream selection
700      */

```

```

779
780     BD_EVENT_AUDIO_STREAM           = 11,
781     BD_EVENT_IG_STREAM              = 12,
782     BD_EVENT_PG_TEXTST_STREAM       = 13,
783     BD_EVENT_PIP_PG_TEXTST_STREAM   = 14,
784     BD_EVENT_SECONDARY_AUDIO_STREAM = 15,
785     BD_EVENT_SECONDARY_VIDEO_STREAM = 16,
787     BD_EVENT_PG_TEXTST              = 17,
788     BD_EVENT_PIP_PG_TEXTST          = 18,
789     BD_EVENT_SECONDARY_AUDIO         = 19,
790     BD_EVENT_SECONDARY_VIDEO         = 20,
791     BD_EVENT_SECONDARY_VIDEO_SIZE    = 21,
793     /*
794      * playback control
795      */
796
797     BD_EVENT_PLAYLIST_STOP           = 22,
799
801     BD_EVENT_DISCONTINUITY           = 23,
804     BD_EVENT_SEEK                    = 24,
807     BD_EVENT_STILL                   = 25,
811     BD_EVENT_STILL_TIME              = 26,
814     BD_EVENT_SOUND_EFFECT            = 27,
816     /*
817      * status
818      */
819
822     BD_EVENT_IDLE                    = 28,
823
825     BD_EVENT_POPUP                   = 29,
828     BD_EVENT_MENU                    = 30,
831     BD_EVENT_STEREOSCOPIC_STATUS     = 31,
834     BD_EVENT_KEY_INTEREST_TABLE     = 32,
837     BD_EVENT_UO_MASK_CHANGED         = 33,
839     /*BD_EVENT_LAST = 33, */
840
841 } bd_event_e;
842
843 typedef struct {
844     uint32_t    event;
845     uint32_t    param;
846 } BD_EVENT;
847
848 /* BD_EVENT_ERROR param values */
849 #define BD_ERROR_HDMV    1
850 #define BD_ERROR_BDJ     2
851 /* BD_EVENT_ENCRYPTED param vlues */
852 #define BD_ERROR_AACS    3
853 #define BD_ERROR_BDPLUS  4
854 /* BD_EVENT_TITLE special titles */
855 #define BLURAY_TITLE_FIRST_PLAY  0xffff
856 #define BLURAY_TITLE_TOP_MENU    0
857 /* BD_EVENT_KEY_INTEREST flags */
858 #define BLURAY_KIT_PLAY          0x1
859 #define BLURAY_KIT_STOP          0x2
860 #define BLURAY_KIT_FFW           0x4
861 #define BLURAY_KIT_REW           0x8
862 #define BLURAY_KIT_TRACK_NEXT    0x10
863 #define BLURAY_KIT_TRACK_PREV    0x20
864 #define BLURAY_KIT_PAUSE         0x40
865 #define BLURAY_KIT_STILL_OFF     0x80
866 #define BLURAY_KIT_SEC_AUDIO     0x100
867 #define BLURAY_KIT_SEC_VIDEO     0x200
868 #define BLURAY_KIT_PG_TEXTST     0x400
869 /* BD_EVENT_UO_MASK flags */
870 #define BLURAY_UO_MENU_CALL      0x1
871 #define BLURAY_UO_TITLE_SEARCH   0x2
872
873 int bd_get_event(BLURAY *bd, BD_EVENT *event);
874
875 const char *bd_event_name(uint32_t /* bd_event_e */ event);
876
877 /*
878  * On-screen display
879  */
880
881 struct bd_overlay_s; /* defined in overlay.h */
882 struct bd_argb_overlay_s; /* defined in overlay.h */
883 struct bd_argb_buffer_s; /* defined in overlay.h */
884
885 typedef void (*bd_overlay_proc_f)(void *handle, const struct bd_overlay_s * const event);
886

```

```

919 typedef void (*bd_argb_overlay_proc_f)(void *handle, const struct bd_argb_overlay_s * const event);
920
940 void bd_register_overlay_proc(BLURAY *bd, void *handle, bd_overlay_proc_f func);
941
957 void bd_register_argb_overlay_proc(BLURAY *bd, void *handle, bd_argb_overlay_proc_f func, struct
    bd_argb_buffer_s *buf);
958
959
960 /*
961  * Playback with on-disc menus
962  */
963
973 int bd_play(BLURAY *bd);
974
987 int bd_play_title(BLURAY *bd, unsigned title);
988
999 int bd_menu_call(BLURAY *bd, int64_t pts);
1000
1013 int bd_read_ext(BLURAY *bd, unsigned char *buf, int len, BD_EVENT *event);
1014
1022 int bd_read_skip_still(BLURAY *bd);
1023
1033 BLURAY_TITLE_INFO* bd_get_playlist_info(BLURAY *bd, uint32_t playlist, unsigned angle);
1034
1044 int bd_get_sound_effect(BLURAY *bd, unsigned sound_id, struct bd_sound_effect *effect);
1045
1046
1047 /*
1048  * User interaction
1049  */
1050
1058 void bd_set_scr(BLURAY *bd, int64_t pts);
1059
1073 int bd_set_rate(BLURAY *bd, uint32_t rate);
1074
1075 #define BLURAY_RATE_PAUSED 0
1076 #define BLURAY_RATE_NORMAL 90000
1093 int bd_user_input(BLURAY *bd, int64_t pts, uint32_t key);
1094
1107 int bd_mouse_select(BLURAY *bd, int64_t pts, uint16_t x, uint16_t y);
1108
1109
1110 /*
1111  * Testing and debugging
1112  *
1113  * Note: parsing functions can't be used with UDF images.
1114  */
1115
1116 /* access to internal information */
1117
1118 struct clpi_cl;
1119 struct mpl_s_pl;
1120 struct bdjo_data;
1121 struct mobj_objects;
1122
1131 struct clpi_cl *bd_get_clpi(BLURAY *bd, unsigned clip_ref);
1132
1134 struct clpi_cl *bd_read_clpi(const char *clpi_file);
1135
1142 void bd_free_clpi(struct clpi_cl *cl);
1143
1144
1146 struct mpl_s_pl *bd_read_mpls(const char *mpls_file);
1148 void bd_free_mpls(struct mpl_s_pl *);
1149
1151 struct mobj_objects *bd_read_mobj(const char *mobj_file);
1153 void bd_free_mobj(struct mobj_objects *);
1154
1156 struct bdjo_data *bd_read_bdjo(const char *bdjo_file);
1158 void bd_free_bdjo(struct bdjo_data *);
1159
1160 /* BD-J testing */
1161
1163 int bd_start_bdj(BLURAY *bd, const char* start_object);
1165 void bd_stop_bdj(BLURAY *bd);
1166
1180 int bd_read_file(BLURAY *bd, const char *path, void **data, int64_t *size);
1181
1192 struct bd_dir_s *bd_open_dir(BLURAY *bd, const char *dir);
1193
1211 struct bd_file_s *bd_open_file_dec(BLURAY *bd, const char *path);

```



```

1212
1213
1214 #ifdef __cplusplus
1215 }
1216 #endif
1217
1218 #endif /* BLURAY_H_ */

```

## 5.9 overlay.h File Reference

Graphics overlay events.

```
#include <stdint.h>
```

### Data Structures

- struct [BD\\_PG\\_PALETTE\\_ENTRY](#)  
*Overlay palette entry.*
- struct [BD\\_PG\\_RLE\\_ELEM](#)  
*RLE element.*
- struct [BD\\_OVERLAY](#)  
*YUV overlay event.*
- struct [BD\\_ARGB\\_OVERLAY](#)  
*ARGB overlay event.*
- struct [BD\\_ARGB\\_BUFFER](#)  
*Application-allocated frame buffer for ARGB overlays.*

### Macros

- #define [BD\\_OVERLAY\\_INTERFACE\\_VERSION](#) 2  
*Version number of the interface described in this file.*

### Enumerations

- enum [bd\\_overlay\\_plane\\_e](#) {  
  [BD\\_OVERLAY\\_PG](#) = 0 ,  
  [BD\\_OVERLAY\\_IG](#) = 1 }  
*Overlay plane.*
- enum [bd\\_overlay\\_cmd\\_e](#) {  
  [BD\\_OVERLAY\\_INIT](#) = 0 ,  
  [BD\\_OVERLAY\\_CLOSE](#) = 1 ,  
  [BD\\_OVERLAY\\_CLEAR](#) = 2 ,  
  [BD\\_OVERLAY\\_DRAW](#) = 3 ,  
  [BD\\_OVERLAY\\_WIPE](#) = 4 ,  
  [BD\\_OVERLAY\\_HIDE](#) = 5 ,  
  [BD\\_OVERLAY\\_FLUSH](#) = 6 }

*YUV overlay event type.*

- enum `bd_argb_overlay_cmd_e` {  
`BD_ARGB_OVERLAY_INIT` = 0 ,  
`BD_ARGB_OVERLAY_CLOSE` = 1 ,  
`BD_ARGB_OVERLAY_DRAW` = 3 ,  
`BD_ARGB_OVERLAY_FLUSH` = 6 }

*ARGB overlay event type.*

## Functions

- const void \* `bd_refcnt_inc` (const void \*)  
*Hold reference-counted object.*
- void `bd_refcnt_dec` (const void \*)  
*Release reference-counted object.*

### 5.9.1 Detailed Description

Graphics overlay events.

### 5.9.2 Enumeration Type Documentation

#### 5.9.2.1 `bd_argb_overlay_cmd_e`

enum `bd_argb_overlay_cmd_e`

ARGB overlay event type.

Enumerator

<code>BD_ARGB_OVERLAY_INIT</code>	Initialize overlay plane. Size and position of plane are in x,y,w,h
<code>BD_ARGB_OVERLAY_CLOSE</code>	Close overlay plane.
<code>BD_ARGB_OVERLAY_DRAW</code>	Draw ARGB image on plane.
<code>BD_ARGB_OVERLAY_FLUSH</code>	All changes have been done, flush overlay to display at given pts.

#### 5.9.2.2 `bd_overlay_cmd_e`

enum `bd_overlay_cmd_e`

YUV overlay event type.

## Enumerator

BD_OVERLAY_INIT	Initialize overlay plane. Size and position of plane in x,y,w,h.
BD_OVERLAY_CLOSE	Close overlay plane.
BD_OVERLAY_CLEAR	Clear overlay plane.
BD_OVERLAY_DRAW	Draw bitmap. Size and position within plane (x, y, w, h) and image (img, palette).
BD_OVERLAY_WIPE	Clear area. Size and position within plane (x, y, w, h).
BD_OVERLAY_HIDE	Overlay is empty and can be hidden.
BD_OVERLAY_FLUSH	All changes have been done, flush overlay to display at given pts.

## 5.9.2.3 bd\_overlay\_plane\_e

```
enum bd_overlay_plane_e
```

Overlay plane.

## Enumerator

BD_OVERLAY_PG	Presentation Graphics plane.
BD_OVERLAY_IG	Interactive Graphics plane (on top of PG plane)

## 5.9.3 Function Documentation

## 5.9.3.1 bd\_refcnt\_inc()

```
const void * bd_refcnt_inc (
    const void * )
```

Hold reference-counted object.

Return object or NULL on invalid object.

## 5.10 overlay.h

[Go to the documentation of this file.](#)

```
1 /*
2  * This file is part of libbluray
3  * Copyright (C) 2010-2017 Petri Hintukainen <phintuka@users.sourceforge.net>
4  *
```

```

5  * This library is free software; you can redistribute it and/or
6  * modify it under the terms of the GNU Lesser General Public
7  * License as published by the Free Software Foundation; either
8  * version 2.1 of the License, or (at your option) any later version.
9  *
10 * This library is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
13 * Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public
16 * License along with this library. If not, see
17 * <http://www.gnu.org/licenses/>.
18 */
19
25 #ifndef BD_OVERLAY_H_
26 #define BD_OVERLAY_H_
27
28 #ifdef __cplusplus
29 extern "C" {
30 #endif
31
32 #include <stdint.h>
33
35 #define BD_OVERLAY_INTERFACE_VERSION 2
36
40 typedef enum {
41     BD_OVERLAY_PG = 0,
42     BD_OVERLAY_IG = 1,
43 } bd_overlay_plane_e;
44
45 /*
46  * Compressed YUV overlays
47  */
48
52 typedef enum {
53     /* following events are executed immediately */
54     BD_OVERLAY_INIT = 0,
55     BD_OVERLAY_CLOSE = 1,
56     /* following events can be processed immediately, but changes
57      * should not be flushed to display before next FLUSH event
58      */
59     BD_OVERLAY_CLEAR = 2,
60     BD_OVERLAY_DRAW = 3,
61     BD_OVERLAY_WIPE = 4,
62     BD_OVERLAY_HIDE = 5,
63     BD_OVERLAY_FLUSH = 6,
64 } bd_overlay_cmd_e;
65
67 typedef struct bd_pg_palette_entry_s {
68     uint8_t Y;
69     uint8_t Cr;
70     uint8_t Cb;
71     uint8_t T;
72 } BD_PG_PALETTE_ENTRY;
73
77 typedef struct bd_pg_rle_elem_s {
78     uint16_t len;
79     uint16_t color;
80 } BD_PG_RLE_ELEM;
81
91
95 typedef struct bd_overlay_s {
96     int64_t pts;
97     uint8_t plane;
98     uint8_t cmd;
99     uint8_t palette_update_flag;
100     uint16_t x;
101     uint16_t y;
102     uint16_t w;
103     uint16_t h;
104     const BD_PG_PALETTE_ENTRY * palette;
105     const BD_PG_RLE_ELEM * img;
106 } BD_OVERLAY;
107
108 /*
109  * RLE images are reference-counted. If application caches rle data for later use,
110  * it needs to use bd_refcnt_inc() and bd_refcnt_dec().
111  */
112
113 const void *bd_refcnt_inc(const void *);
114 void bd_refcnt_dec(const void *);

```

```

120 #if 0
121 BD_OVERLAY *bd_overlay_copy(const BD_OVERLAY *src)
122 {
123     BD_OVERLAY *ov = malloc(sizeof(*ov));
124     memcpy(ov, src, sizeof(*ov));
125     if (ov->palette) {
126         ov->palette = malloc(256 * sizeof(BD_PG_PALETTE_ENTRY));
127         memcpy((void*)ov->palette, src->palette, 256 * sizeof(BD_PG_PALETTE_ENTRY));
128     }
129     if (ov->img) {
130         bd_refcnt_inc(ov->img);
131     }
132     return ov;
133 }
134
135 void bd_overlay_free(BD_OVERLAY **pov)
136 {
137     if (pov && *pov) {
138         BD_OVERLAY *ov = *pov;
139         void *p = (void*)ov->palette;
140         bd_refcnt_dec(ov->img);
141         X_FREE(p);
142         ov->palette = NULL;
143         X_FREE(*pov);
144     }
145 }
146 #endif
147
148 typedef enum {
149     /* following events are executed immediately */
150     BD_ARGB_OVERLAY_INIT = 0,
151     BD_ARGB_OVERLAY_CLOSE = 1,
152     /* following events can be processed immediately, but changes
153      * should not be flushed to display before next FLUSH event
154      */
155     BD_ARGB_OVERLAY_DRAW = 3,
156     BD_ARGB_OVERLAY_FLUSH = 6,
157 } bd_argb_overlay_cmd_e;
158
159 typedef struct bd_argb_overlay_s {
160     int64_t pts;
161     uint8_t plane;
162     uint8_t cmd;
163     /* following fields are used only when not using application-allocated
164      * frame buffer
165      */
166     /* destination clip on the overlay plane */
167     uint16_t x;
168     uint16_t y;
169     uint16_t w;
170     uint16_t h;
171     uint16_t stride;
172     const uint32_t * argb;
173 } BD_ARGB_OVERLAY;
174
175 typedef struct bd_argb_buffer_s {
176     /* optional lock / unlock functions
177      * - Set by application
178      * - Called when buffer is accessed or modified
179      */
180     void (*lock) (struct bd_argb_buffer_s *);
181     void (*unlock) (struct bd_argb_buffer_s *);
182     /* ARGB frame buffers
183      * - Allocated by application (BD_ARGB_OVERLAY_INIT).
184      * - Buffer can be freed after BD_ARGB_OVERLAY_CLOSE.
185      * - buffer can be replaced in overlay callback or lock().
186      */
187     uint32_t *buf[4];
188     /* size of buffers
189      * - Set by application
190      * - If the buffer size is smaller than the size requested in BD_ARGB_OVERLAY_INIT,
191      *   the buffer points only to the dirty area.
192      */
193     int width;
194     int height;
195     struct {
196         uint16_t x0;
197         uint16_t y0;
198         uint16_t x1;
199     };

```

```

227         uint16_t y1;
228     } dirty[2];
229 } BD_ARGB_BUFFER;
230
231 #ifdef __cplusplus
232 }
233 #endif
234
235 #endif // BD_OVERLAY_H_

```

## 5.11 keys.h File Reference

User input key definitions.

### Macros

- `#define BD_VK_KEY_PRESSED 0x80000000`  
*Key was pressed down.*
- `#define BD_VK_KEY_TYPED 0x40000000`  
*Key was typed.*
- `#define BD_VK_KEY_RELEASED 0x20000000`  
*Key was released.*

### Enumerations

- `enum bd_vk_key_e {`  
`BD_VK_NONE = 0xffff ,`  
`BD_VK_0 = 0 ,`  
`BD_VK_1 = 1 ,`  
`BD_VK_2 = 2 ,`  
`BD_VK_3 = 3 ,`  
`BD_VK_4 = 4 ,`  
`BD_VK_5 = 5 ,`  
`BD_VK_6 = 6 ,`  
`BD_VK_7 = 7 ,`  
`BD_VK_8 = 8 ,`  
`BD_VK_9 = 9 ,`  
`BD_VK_ROOT_MENU = 10 ,`  
`BD_VK_POPUP = 11 ,`  
`BD_VK_UP = 12 ,`  
`BD_VK_DOWN = 13 ,`  
`BD_VK_LEFT = 14 ,`  
`BD_VK_RIGHT = 15 ,`  
`BD_VK_ENTER = 16 ,`  
`BD_VK_MOUSE_ACTIVATE = 17 ,`  
`BD_VK_RED = 403 ,`  
`BD_VK_GREEN = 404 ,`  
`BD_VL_YELLOW = 405 ,`  
`BD_VK_BLUE = 406 }`  
*Key codes.*

### 5.11.1 Detailed Description

User input key definitions.

### 5.11.2 Enumeration Type Documentation

#### 5.11.2.1 bd\_vk\_key\_e

```
enum bd_vk_key_e
```

Key codes.

Enumerator

BD_VK_NONE	no key pressed
BD_VK_0	"1"
BD_VK_1	"2"
BD_VK_2	"3"
BD_VK_3	"4"
BD_VK_4	"5"
BD_VK_5	"6"
BD_VK_6	"7"
BD_VK_7	"8"
BD_VK_8	"9"
BD_VK_9	"0"
BD_VK_ROOT_MENU	Open disc root menu.
BD_VK_POPUP	Toggle popup menu.
BD_VK_UP	Arrow up.
BD_VK_DOWN	Arrow down.
BD_VK_LEFT	Arrow left.
BD_VK_RIGHT	Arrow right.
BD_VK_ENTER	Select.
BD_VK_MOUSE_ACTIVATE	Mouse click. Translated to BD_VK_ENTER if mouse is over a valid button.
BD_VK_RED	Color key "Red".
BD_VK_GREEN	Color key "Green".
BD_VL_YELLOW	Color key "Yellow".
BD_VK_BLUE	Color key "Blue".

## 5.12 keys.h

[Go to the documentation of this file.](#)

```

1 /*
2  * This file is part of libbluray
3  * Copyright (C) 2010-2022 Petri Hintukainen <phintuka@users.sourceforge.net>
4  *
5  * This library is free software; you can redistribute it and/or
6  * modify it under the terms of the GNU Lesser General Public
7  * License as published by the Free Software Foundation; either
8  * version 2.1 of the License, or (at your option) any later version.
9  *
10 * This library is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
13 * Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public
16 * License along with this library. If not, see
17 * <http://www.gnu.org/licenses/>.
18 */
19
25 #if !defined(_BD_KEYS_H_)
26 #define _BD_KEYS_H_
27
31 typedef enum {
32     BD_VK_NONE          = 0xffff,
33     /* numeric key events */
34     BD_VK_0             = 0,
35     BD_VK_1             = 1,
36     BD_VK_2             = 2,
37     BD_VK_3             = 3,
38     BD_VK_4             = 4,
39     BD_VK_5             = 5,
40     BD_VK_6             = 6,
41     BD_VK_7             = 7,
42     BD_VK_8             = 8,
43     BD_VK_9             = 9,
44     /* */
45     BD_VK_ROOT_MENU    = 10,
46     BD_VK_POPUP        = 11,
47     /* interactive key events */
48     BD_VK_UP           = 12,
49     BD_VK_DOWN         = 13,
50     BD_VK_LEFT         = 14,
51     BD_VK_RIGHT        = 15,
52     BD_VK_ENTER        = 16,
53     BD_VK_MOUSE_ACTIVATE = 17,
54
55     BD_VK_RED           = 403,
56     BD_VK_GREEN         = 404,
57     BD_VL_YELLOW        = 405,
58     BD_VK_BLUE          = 406,
59 } bd_vk_key_e;
60
61 /*
62 * Application may optionally provide KEY_PRESSED, KEY_TYPED and KEY_RELEASED events.
63 * These masks are OR'd with the key code when calling bd_user_input().
64 */
65
66 #define BD_VK_KEY_PRESSED    0x80000000
67 #define BD_VK_KEY_TYPED     0x40000000
68 #define BD_VK_KEY_RELEASED  0x20000000
69 #endif // _BD_KEYS_H_

```

## 5.13 player\_settings.h File Reference

Definitions for Blu-Ray player settings.

### Macros

- **#define BLURAY\_DCAP\_1080p\_720p\_3D 0x01**  
*capable of 1920x1080 23.976Hz and 1280x720 59.94Hz 3D*



- #define **BLURAY\_DCAP\_720p\_50Hz\_3D** 0x02  
*capable of 1280x720 50Hz 3D*
- #define **BLURAY\_DCAP\_NO\_3D\_CLASSES\_REQUIRED** 0x04  
*3D glasses are not required*
- #define **BLURAY\_DCAP\_INTERLACED\_3D** 0x08  
*capable of interlaced 3D*
- #define **BLURAY\_DCAP\_DISPLAY\_SIZE\_UNDEFINED** 0  
*connected display physical size unknown/undefined*
- #define **BLURAY\_DCAP\_DISPLAY\_SIZE\_MASK** 0xffff00  
*display size mask*
- #define **BLURAY\_DCAP\_DISPLAY\_SIZE(cm)** (((cm) > 0xffff ? 0xffff : (cm)) << 8)  
*connected display physical size (cm)*
- #define **BLURAY\_PLAYER\_PROFILE\_3D\_FLAG** 0x100000  
*set for 3D profiles*
- #define **BLURAY\_PLAYER\_PROFILE\_VERSION\_MASK** 0xffff  
*bit mask for player version*

## Enumerations

- enum {  
**BLURAY\_ACAP\_LPCM\_48\_96\_STEREO\_ONLY** = 0x0001 ,  
**BLURAY\_ACAP\_LPCM\_48\_96\_SURROUND** = 0x0002 ,  
**BLURAY\_ACAP\_LPCM\_192\_NONE** = 0x0000 ,  
**BLURAY\_ACAP\_LPCM\_192\_STEREO\_ONLY** = 0x0004 ,  
**BLURAY\_ACAP\_LPCM\_192\_SURROUND** = 0x0008 ,  
**BLURAY\_ACAP\_DDPLUS\_STEREO\_ONLY** = 0x0010 ,  
**BLURAY\_ACAP\_DDPLUS\_SURROUND** = 0x0020 ,  
**BLURAY\_ACAP\_DDPLUS\_DEP\_NONE** = 0x0000 ,  
**BLURAY\_ACAP\_DDPLUS\_DEP\_STEREO\_ONLY** = 0x0040 ,  
**BLURAY\_ACAP\_DDPLUS\_DEP\_SURROUND** = 0x0080 ,  
**BLURAY\_ACAP\_DTSHD\_CORE\_STEREO\_ONLY** = 0x0100 ,  
**BLURAY\_ACAP\_DTSHD\_CORE\_SURROUND** = 0x0200 ,  
**BLURAY\_ACAP\_DTSHD\_EXT\_NONE** = 0x0000 ,  
**BLURAY\_ACAP\_DTSHD\_EXT\_STEREO\_ONLY** = 0x0400 ,  
**BLURAY\_ACAP\_DTSHD\_EXT\_SURROUND** = 0x0800 ,  
**BLURAY\_ACAP\_DD\_STEREO\_ONLY** = 0x1000 ,  
**BLURAY\_ACAP\_DD\_SURROUND** = 0x2000 ,  
**BLURAY\_ACAP\_MLP\_NONE** = 0x0000 ,  
**BLURAY\_ACAP\_MLP\_STEREO\_ONLY** = 0x4000 ,  
**BLURAY\_ACAP\_MLP\_SURROUND** = 0x8000 }  
**BLURAY\_PLAYER\_SETTING\_AUDIO\_CAP (PSR15)**
- enum {  
**BLURAY\_REGION\_A** = 1 ,  
**BLURAY\_REGION\_B** = 2 ,  
**BLURAY\_REGION\_C** = 4 }  
**BLURAY\_PLAYER\_SETTING\_REGION\_CODE (PSR20)**
- enum {  
**BLURAY\_OUTPUT\_PREFER\_2D** = 0 ,  
**BLURAY\_OUTPUT\_PREFER\_3D** = 1 }  
**BLURAY\_PLAYER\_SETTING\_OUTPUT\_PREFER (PSR21)**

- enum {  
`BLURAY_VCAP_SECONDARY_HD` = 0x01 ,  
`BLURAY_VCAP_25Hz_50Hz` = 0x02 }  
*BLURAY\_PLAYER\_SETTING\_VIDEO\_CAP (PSR29)*
- enum {  
`BLURAY_PLAYER_PROFILE_1_v1_0` = ((0x00 << 16) | (0x0100)) ,  
`BLURAY_PLAYER_PROFILE_1_v1_1` = ((0x01 << 16) | (0x0110)) ,  
`BLURAY_PLAYER_PROFILE_2_v2_0` = ((0x03 << 16) | (0x0200)) ,  
`BLURAY_PLAYER_PROFILE_3_v2_0` = ((0x08 << 16) | (0x0200)) ,  
`BLURAY_PLAYER_PROFILE_5_v2_4` = ((0x13 << 16) | (0x0240)) ,  
`BLURAY_PLAYER_PROFILE_6_v3_0` = ((0x00 << 16) | (0x0300)) ,  
`BLURAY_PLAYER_PROFILE_6_v3_1` = ((0x00 << 16) | (0x0310)) }  
*BLURAY\_PLAYER\_SETTING\_PLAYER\_PROFILE (PSR31)*
- enum {  
`BLURAY_PG_TEXTST_DECODER_DISABLE` = 0 ,  
`BLURAY_PG_TEXTST_DECODER_ENABLE` = 1 }  
*BLURAY\_PLAYER\_SETTING\_DECODE\_PG.*
- enum {  
`BLURAY_PERSISTENT_STORAGE_DISABLE` = 0 ,  
`BLURAY_PERSISTENT_STORAGE_ENABLE` = 1 }  
*BLURAY\_PLAYER\_SETTING\_PERSISTENT\_STORAGE.*

### 5.13.1 Detailed Description

Definitions for Blu-Ray player settings.

### 5.13.2 Enumeration Type Documentation

#### 5.13.2.1 anonymous enum

anonymous enum

`BLURAY_PLAYER_SETTING_AUDIO_CAP` (PSR15)

Player capability for audio (bitmask)

Enumerator

<code>BLURAY_ACAP_LPCM_48_96_STEREO_ONLY</code>	LPCM 48kHz and 96kHz stereo capable.
<code>BLURAY_ACAP_LPCM_48_96_SURROUND</code>	LPCM 48kHz and 96kHz surround capable.
<code>BLURAY_ACAP_LPCM_192_NONE</code>	LPCM 192kHz not supported.
<code>BLURAY_ACAP_LPCM_192_STEREO_ONLY</code>	LPCM 192kHz stereo capable.
<code>BLURAY_ACAP_LPCM_192_SURROUND</code>	LPCM 192kHz surround capable.
<code>BLURAY_ACAP_DDPLUS_STEREO_ONLY</code>	DD Plus independent substream stereo capable.

## Enumerator

BLURAY_ACAP_DDPLUS_SURROUND	DD Plus independent substream surround capable.
BLURAY_ACAP_DDPLUS_DEP_NONE	DD Plus dependent substream not supported.
BLURAY_ACAP_DDPLUS_DEP_STEREO_ONLY	DD Plus dependent substream stereo capable.
BLURAY_ACAP_DDPLUS_DEP_SURROUND	DD Plus dependent substream surround capable.
BLURAY_ACAP_DTSHD_CORE_STEREO_ONLY	DTS-HD Core stereo capable.
BLURAY_ACAP_DTSHD_CORE_SURROUND	DTS-HD Core surround capable.
BLURAY_ACAP_DTSHD_EXT_NONE	DTS-HD extension substream not supported.
BLURAY_ACAP_DTSHD_EXT_STEREO_ONLY	DTS-HD extension substream stereo capable.
BLURAY_ACAP_DTSHD_EXT_SURROUND	DTS-HD extension substream surround capable.
BLURAY_ACAP_DD_STEREO_ONLY	Dolby Digital audio stereo capable.
BLURAY_ACAP_DD_SURROUND	Dolby Digital audio surround capable.
BLURAY_ACAP_MLP_NONE	MLP not supported.
BLURAY_ACAP_MLP_STEREO_ONLY	MLP stereo capable.
BLURAY_ACAP_MLP_SURROUND	MLP surround capable.

## 5.13.2.2 anonymous enum

anonymous enum

BLURAY\_PLAYER\_SETTING\_REGION\_CODE (PSR20)

Player region code (integer)

## Enumerator

BLURAY_REGION_A	Region A: the Americas, East and Southeast Asia, U.S. territories, and Bermuda.
BLURAY_REGION_B	Region B: Africa, Europe, Oceania, the Middle East, the Kingdom of the Netherlands, British overseas territories, French territories, and Greenland.
BLURAY_REGION_C	Region C: Central and South Asia, Mongolia, Russia, and the People's Republic of China.

## 5.13.2.3 anonymous enum

anonymous enum

BLURAY\_PLAYER\_SETTING\_OUTPUT\_PREFER (PSR21)

Output mode preference (integer)

## Enumerator

BLURAY_OUTPUT_PREFER_2D	2D output preferred
BLURAY_OUTPUT_PREFER_3D	3D output preferred

**5.13.2.4 anonymous enum**

anonymous enum

BLURAY\_PLAYER\_SETTING\_VIDEO\_CAP (PSR29)

Player capability for video (bit mask)

## Enumerator

BLURAY_VCAP_SECONDARY_HD	player can play secondary stream in HD
BLURAY_VCAP_25Hz_50Hz	player can play 25Hz and 50Hz video

**5.13.2.5 anonymous enum**

anonymous enum

BLURAY\_PLAYER\_SETTING\_PLAYER\_PROFILE (PSR31)

Player profile and version

- Profile 1, version 1.0: no local storage, no VFS, no internet
- Profile 1, version 1.1: PiP, VFS, sec. audio, 256MB local storage, no internet
- Profile 2, version 2.0: BdLive (internet), 1GB local storage

## Enumerator

BLURAY_PLAYER_PROFILE_1_v1↔ _0	Profile 1, version 1.0 (Initial Standard Profile)
BLURAY_PLAYER_PROFILE_1_v1↔ _1	Profile 1, version 1.1 (secondary stream support)
BLURAY_PLAYER_PROFILE_2_v2↔ _0	Profile 2, version 2.0 (network access, BdLive)
BLURAY_PLAYER_PROFILE_3_v2↔ _0	Profile 3, version 2.0 (audio only player)

## Enumerator

BLURAY_PLAYER_PROFILE_5_v2↔ _4	Profile 5, version 2.4 (3D)
BLURAY_PLAYER_PROFILE_6_v3↔ _0	Profile 6, version 3.0 (UHD)
BLURAY_PLAYER_PROFILE_6_v3↔ _1	Profile 6, version 3.1 (UHD)

## 5.13.2.6 anonymous enum

anonymous enum

BLURAY\_PLAYER\_SETTING\_DECODE\_PG.

Enable Presentation Graphics and Text Subtitle decoder

## Enumerator

BLURAY_PG_TEXTST_DECODER_DISABLE	disable both decoders
BLURAY_PG_TEXTST_DECODER_ENABLE	enable both decoders

## 5.13.2.7 anonymous enum

anonymous enum

BLURAY\_PLAYER\_SETTING\_PERSISTENT\_STORAGE.

Enable / disable BD-J persistent storage.

If persistent storage is disabled, BD-J Xlets can't access any data stored during earlier playback sessions. Persistent data stored during current playback session will be removed and can't be accessed later.

This setting can't be changed after [bd\\_play\(\)](#) has been called.

## Enumerator

BLURAY_PERSISTENT_STORAGE_DISABLE	disable persistent storage between playback sessions
BLURAY_PERSISTENT_STORAGE_ENABLE	enable persistent storage

## 5.14 player\_settings.h

[Go to the documentation of this file.](#)

```

1 /*
2  * This file is part of libbluray
3  * Copyright (C) 2014-2017 VideoLAN
4  *
5  * This library is free software; you can redistribute it and/or
6  * modify it under the terms of the GNU Lesser General Public
7  * License as published by the Free Software Foundation; either
8  * version 2.1 of the License, or (at your option) any later version.
9  *
10 * This library is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
13 * Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public
16 * License along with this library. If not, see
17 * <http://www.gnu.org/licenses/>.
18 */
19
20 #ifndef BD_PLAYER_SETTINGS_H_
21 #define BD_PLAYER_SETTINGS_H_
22
23 enum {
24
25     /* LPCM capability */
26
27     /* 48/96kHz (mandatory) */
28     BLURAY_ACAP_LPCM_48_96_STEREO_ONLY = 0x0001,
29     BLURAY_ACAP_LPCM_48_96_SURROUND   = 0x0002,
30
31     /* 192kHz (optional) */
32     BLURAY_ACAP_LPCM_192_NONE         = 0x0000,
33     BLURAY_ACAP_LPCM_192_STEREO_ONLY  = 0x0004,
34     BLURAY_ACAP_LPCM_192_SURROUND     = 0x0008,
35
36     /* Dolby Digital Plus capability */
37
38     /* independent substream (mandatory) */
39     BLURAY_ACAP_DDPLUS_STEREO_ONLY   = 0x0010,
40     BLURAY_ACAP_DDPLUS_SURROUND      = 0x0020,
41
42     /* dependent substream (optional) */
43     BLURAY_ACAP_DDPLUS_DEP_NONE      = 0x0000,
44     BLURAY_ACAP_DDPLUS_DEP_STEREO_ONLY = 0x0040,
45     BLURAY_ACAP_DDPLUS_DEP_SURROUND   = 0x0080,
46
47     /* DTS-HD */
48
49     /* Core substream (mandatory) */
50     BLURAY_ACAP_DTSHD_CORE_STEREO_ONLY = 0x0100,
51     BLURAY_ACAP_DTSHD_CORE_SURROUND   = 0x0200,
52
53     /* Extension substream (optional) */
54     BLURAY_ACAP_DTSHD_EXT_NONE        = 0x0000,
55     BLURAY_ACAP_DTSHD_EXT_STEREO_ONLY = 0x0400,
56     BLURAY_ACAP_DTSHD_EXT_SURROUND    = 0x0800,
57
58     /* Dolby lossless (TrueHD) */
59
60     /* Dolby Digital (mandatory) */
61     BLURAY_ACAP_DD_STEREO_ONLY        = 0x1000,
62     BLURAY_ACAP_DD_SURROUND           = 0x2000,
63
64     /* MLP (optional) */
65     BLURAY_ACAP_MLP_NONE              = 0x0000,
66     BLURAY_ACAP_MLP_STEREO_ONLY       = 0x4000,
67     BLURAY_ACAP_MLP_SURROUND          = 0x8000,
68 };
69
70 enum {
71     BLURAY_REGION_A = 1,
72
73     BLURAY_REGION_B = 2,
74     BLURAY_REGION_C = 4,
75 };
76
77 enum {
78     BLURAY_OUTPUT_PREFER_2D = 0,
79     BLURAY_OUTPUT_PREFER_3D = 1,
80 };
81
82

```

```

112
113 /*
114  * BLURAY_PLAYER_SETTING_DISPLAY_CAP (PSR23)
115  *
116  * Display capability (bit mask) and display size
117  */
118
119 #define BLURAY_DCAP_1080p_720p_3D          0x01
120 #define BLURAY_DCAP_720p_50Hz_3D          0x02
121 #define BLURAY_DCAP_NO_3D_CLASSES_REQUIRED 0x04
122 #define BLURAY_DCAP_INTERLACED_3D         0x08
123 /* horizontal display size in centimeters */
124 #define BLURAY_DCAP_DISPLAY_SIZE_UNDEFINED 0
125 #define BLURAY_DCAP_DISPLAY_SIZE_MASK     0xffff00
126 #define BLURAY_DCAP_DISPLAY_SIZE(cm)      (((cm) > 0xffff ? 0xffff : (cm)) << 8)
127
128 enum {
129     BLURAY_VCAP_SECONDARY_HD = 0x01,
130     BLURAY_VCAP_25Hz_50Hz   = 0x02,
131 };
132
133 enum {
134     BLURAY_PLAYER_PROFILE_1_v1_0 = ((0x00 << 16) | (0x0100)),
135     BLURAY_PLAYER_PROFILE_1_v1_1 = ((0x01 << 16) | (0x0110)),
136     BLURAY_PLAYER_PROFILE_2_v2_0 = ((0x03 << 16) | (0x0200)),
137     BLURAY_PLAYER_PROFILE_3_v2_0 = ((0x08 << 16) | (0x0200)),
138     BLURAY_PLAYER_PROFILE_5_v2_4 = ((0x13 << 16) | (0x0240)),
139     BLURAY_PLAYER_PROFILE_6_v3_0 = ((0x00 << 16) | (0x0300)),
140     BLURAY_PLAYER_PROFILE_6_v3_1 = ((0x00 << 16) | (0x0310)),
141 };
142
143 /* Player profile flags and version mask */
144
145 #define BLURAY_PLAYER_PROFILE_3D_FLAG      0x100000
146 #define BLURAY_PLAYER_PROFILE_VERSION_MASK 0xffff
147
148 enum {
149     BLURAY_PG_TEXTST_DECODER_DISABLE = 0,
150     BLURAY_PG_TEXTST_DECODER_ENABLE  = 1,
151 };
152
153 enum {
154     BLURAY_PERSISTENT_STORAGE_DISABLE = 0,
155     BLURAY_PERSISTENT_STORAGE_ENABLE  = 1,
156 };
157
158 #endif /* BD_PLAYER_SETTINGS_H */

```

## 5.15 log\_control.h File Reference

Log control and capture.

```
#include <stdint.h>
```

### Typedefs

- typedef void(\* [BD\\_LOG\\_FUNC](#)) (const char \*msg)

*Log a message.*

## Enumerations

- enum `debug_mask_t` {  
`DBG_RESERVED` = 0x00001 ,  
`DBG_CONFIGFILE` = 0x00002 ,  
`DBG_FILE` = 0x00004 ,  
`DBG_AACS` = 0x00008 ,  
`DBG_MKB` = 0x00010 ,  
`DBG_MMC` = 0x00020 ,  
`DBG_BLURAY` = 0x00040 ,  
`DBG_DIR` = 0x00080 ,  
`DBG_NAV` = 0x00100 ,  
`DBG_BDPLUS` = 0x00200 ,  
`DBG_DLX` = 0x00400 ,  
`DBG_CRIT` = 0x00800 ,  
`DBG_HDMV` = 0x01000 ,  
`DBG_BDJ` = 0x02000 ,  
`DBG_STREAM` = 0x04000 ,  
`DBG_GC` = 0x08000 ,  
`DBG_DECODE` = 0x10000 ,  
`DBG_JNI` = 0x20000 }

*Flags for log filtering.*

## Functions

- void `bd_set_debug_handler` (`BD_LOG_FUNC` handler)  
*Set (global) debug handler.*
- void `bd_set_debug_mask` (uint32\_t mask)  
*Set (global) debug mask.*
- uint32\_t `bd_get_debug_mask` (void)  
*Get current (global) debug mask.*

### 5.15.1 Detailed Description

Log control and capture.

Logging level can be changed with function `bd_set_debug_mask()` or environment variable `BD_DEBUG_MASK`. Default is to log only errors and critical messages (`DBG_CRIT`).

Application can capture log messages with `bd_set_debug_handler()`. Messages can be written to a log file with `BD_DEBUG_FILE` environment variable. By default messages are written to standard error output.

### 5.15.2 Typedef Documentation

#### 5.15.2.1 BD\_LOG\_FUNC

```
typedef void(* BD_LOG_FUNC) (const char *msg)
```

Log a message.



## Parameters

<i>msg</i>	Log message as null-terminated string
------------	---------------------------------------

### 5.15.3 Enumeration Type Documentation

#### 5.15.3.1 debug\_mask\_t

enum [debug\\_mask\\_t](#)

Flags for log filtering.

## Enumerator

DBG_BLURAY	BluRay player.
DBG_DIR	Directory access.
DBG_NAV	Database files (playlist and clip info)
DBG_CRIT	<b>Critical messages and errors</b> (default)
DBG_HDMV	HDMV virtual machine execution trace.
DBG_BDJ	BD-J subsystem and Xlet trace.
DBG_STREAM	m2ts stream trace
DBG_GC	graphics controller trace
DBG_DECODE	PG / IG decoders, m2ts demuxer.
DBG_JNI	JNI calls.

### 5.15.4 Function Documentation

#### 5.15.4.1 bd\_get\_debug\_mask()

```
uint32_t bd_get_debug_mask (  
    void )
```

Get current (global) debug mask.

## Returns

combination of flags from debug\_mask\_enum

#### 5.15.4.2 `bd_set_debug_handler()`

```
void bd_set_debug_handler (
    BD_LOG_FUNC handler )
```

Set (global) debug handler.

The function will receive all enabled log messages.

##### Parameters

<i>handler</i>	function that will receive all enabled log and trace messages
----------------	---

#### 5.15.4.3 `bd_set_debug_mask()`

```
void bd_set_debug_mask (
    uint32_t mask )
```

Set (global) debug mask.

##### Parameters

<i>mask</i>	combination of flags from <code>debug_mask_enum</code>
-------------	--

## 5.16 `log_control.h`

[Go to the documentation of this file.](#)

```
1 /*
2  * This file is part of libbluray
3  * Copyright (C) 2009-2010  Obliter0n
4  * Copyright (C) 2009-2010  John Stebbins
5  *
6  * This library is free software; you can redistribute it and/or
7  * modify it under the terms of the GNU Lesser General Public
8  * License as published by the Free Software Foundation; either
9  * version 2.1 of the License, or (at your option) any later version.
10 *
11 * This library is distributed in the hope that it will be useful,
12 * but WITHOUT ANY WARRANTY; without even the implied warranty of
13 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
14 * Lesser General Public License for more details.
15 *
16 * You should have received a copy of the GNU Lesser General Public
17 * License along with this library. If not, see
18 * <http://www.gnu.org/licenses/>.
19 */
20
23 #ifndef BD_LOG_CONTROL_H_
24 #define BD_LOG_CONTROL_H_
25
26 #ifdef __cplusplus
27 extern "C" {
```

```
38 #endif
39
40 #include <stdint.h>
41
42 typedef enum debug_mask_enum {
43     DBG_RESERVED = 0x00001, /* (reserved) */
44     DBG_CONFIGFILE = 0x00002, /* (reserved for libaacs) */
45     DBG_FILE = 0x00004, /* (reserved for libaacs) */
46     DBG_AACS = 0x00008, /* (reserved for libaacs) */
47     DBG_MKB = 0x00010, /* (reserved for libaacs) */
48     DBG_MMC = 0x00020, /* (reserved for libaacs) */
49     DBG_BLURAY = 0x00040,
50     DBG_DIR = 0x00080,
51     DBG_NAV = 0x00100,
52     DBG_BDPLUS = 0x00200, /* (reserved for libbdplus) */
53     DBG_DLX = 0x00400, /* (reserved for libbdplus) */
54     DBG_CRIT = 0x00800,
55     DBG_HDMV = 0x01000,
56     DBG_BDJ = 0x02000,
57     DBG_STREAM = 0x04000,
58     DBG_GC = 0x08000,
59     DBG_DECODE = 0x10000,
60     DBG_JNI = 0x20000,
61 } debug_mask_t;
62
63 typedef void (*BD_LOG_FUNC) (const char *msg);
64
65 void bd_set_debug_handler(BD_LOG_FUNC handler);
66
67 void bd_set_debug_mask(uint32_t mask);
68
69 uint32_t bd_get_debug_mask(void);
70
71 #ifdef __cplusplus
72 }
73 #endif
74
75 #endif /* BD_LOG_CONTROL_H_ */
```



# Index

BD\_ARGB\_BUFFER, [7](#)  
    buf, [8](#)  
    dirty, [8](#)  
BD\_ARGB\_OVERLAY, [8](#)  
BD\_ARGB\_OVERLAY\_CLOSE  
    overlay.h, [82](#)  
bd\_argb\_overlay\_cmd\_e  
    overlay.h, [82](#)  
BD\_ARGB\_OVERLAY\_DRAW  
    overlay.h, [82](#)  
BD\_ARGB\_OVERLAY\_FLUSH  
    overlay.h, [82](#)  
BD\_ARGB\_OVERLAY\_INIT  
    overlay.h, [82](#)  
bd\_argb\_overlay\_proc\_f  
    bluray.h, [44](#)  
bd\_chapter\_pos  
    bluray.h, [48](#)  
bd\_close  
    bluray.h, [49](#)  
BD\_DIR\_H, [9](#)  
    close, [10](#)  
    read, [10](#)  
BD\_DIR\_OPEN  
    filesystem.h, [30](#)  
BD\_DIRENT, [10](#)  
BD\_EVENT, [11](#)  
BD\_EVENT\_ANGLE  
    bluray.h, [45](#)  
BD\_EVENT\_AUDIO\_STREAM  
    bluray.h, [45](#)  
BD\_EVENT\_CHAPTER  
    bluray.h, [45](#)  
BD\_EVENT\_DISCONTINUITY  
    bluray.h, [45](#)  
bd\_event\_e  
    bluray.h, [45](#)  
BD\_EVENT\_ENCRYPTED  
    bluray.h, [45](#)  
BD\_EVENT\_END\_OF\_TITLE  
    bluray.h, [45](#)  
BD\_EVENT\_ERROR  
    bluray.h, [45](#)  
BD\_EVENT\_IDLE  
    bluray.h, [46](#)  
BD\_EVENT\_IG\_STREAM  
    bluray.h, [45](#)  
BD\_EVENT\_KEY\_INTEREST\_TABLE  
    bluray.h, [46](#)  
BD\_EVENT\_MENU  
    bluray.h, [46](#)  
bd\_event\_name  
    bluray.h, [49](#)  
BD\_EVENT\_NONE  
    bluray.h, [45](#)  
BD\_EVENT\_PG\_TEXTST  
    bluray.h, [45](#)  
BD\_EVENT\_PG\_TEXTST\_STREAM  
    bluray.h, [45](#)  
BD\_EVENT\_PIP\_PG\_TEXTST  
    bluray.h, [45](#)  
BD\_EVENT\_PIP\_PG\_TEXTST\_STREAM  
    bluray.h, [45](#)  
BD\_EVENT\_PLAYITEM  
    bluray.h, [45](#)  
BD\_EVENT\_PLAYLIST  
    bluray.h, [45](#)  
BD\_EVENT\_PLAYLIST\_STOP  
    bluray.h, [45](#)  
BD\_EVENT\_PLAYMARK  
    bluray.h, [45](#)  
BD\_EVENT\_POPUP  
    bluray.h, [46](#)  
BD\_EVENT\_READ\_ERROR  
    bluray.h, [45](#)  
BD\_EVENT\_SECONDARY\_AUDIO  
    bluray.h, [45](#)  
BD\_EVENT\_SECONDARY\_AUDIO\_STREAM  
    bluray.h, [45](#)  
BD\_EVENT\_SECONDARY\_VIDEO  
    bluray.h, [45](#)  
BD\_EVENT\_SECONDARY\_VIDEO\_SIZE  
    bluray.h, [45](#)  
BD\_EVENT\_SECONDARY\_VIDEO\_STREAM  
    bluray.h, [45](#)  
BD\_EVENT\_SEEK  
    bluray.h, [45](#)  
BD\_EVENT\_SOUND\_EFFECT  
    bluray.h, [45](#)  
BD\_EVENT\_STEREOSCOPIC\_STATUS

- bluray.h, 46
- BD\_EVENT\_STILL
  - bluray.h, 45
- BD\_EVENT\_STILL\_TIME
  - bluray.h, 45
- BD\_EVENT\_TITLE
  - bluray.h, 45
- BD\_EVENT\_UO\_MASK\_CHANGED
  - bluray.h, 46
- BD\_FILE\_H, 11
  - close, 12
  - eof, 12
  - internal, 13
  - read, 13
  - seek, 13
  - tell, 14
  - write, 14
- BD\_FILE\_OPEN
  - filesystem.h, 30
- bd\_free\_clpi
  - bluray.h, 49
- bd\_free\_title\_info
  - bluray.h, 50
- bd\_get\_clpi
  - bluray.h, 50
- bd\_get\_current\_angle
  - bluray.h, 50
- bd\_get\_current\_chapter
  - bluray.h, 51
- bd\_get\_current\_title
  - bluray.h, 51
- bd\_get\_debug\_mask
  - log\_control.h, 97
- bd\_get\_disc\_info
  - bluray.h, 51
- bd\_get\_event
  - bluray.h, 52
- bd\_get\_main\_title
  - bluray.h, 52
- bd\_get\_meta
  - bluray.h, 53
- bd\_get\_meta\_file
  - bluray.h, 53
- bd\_get\_playlist\_info
  - bluray.h, 54
- bd\_get\_sound\_effect
  - bluray.h, 54
- bd\_get\_title\_info
  - bluray.h, 55
- bd\_get\_title\_size
  - bluray.h, 55
- bd\_get\_titles
  - bluray.h, 56
- bd\_get\_version

- bluray.h, 56
- bd\_init
  - bluray.h, 57
- BD\_LOG\_FUNC
  - log\_control.h, 96
- bd\_mark\_type\_e
  - bluray.h, 46
- bd\_menu\_call
  - bluray.h, 57
- bd\_mouse\_select
  - bluray.h, 57
- bd\_open
  - bluray.h, 58
- bd\_open\_dir
  - bluray.h, 58
- bd\_open\_disc
  - bluray.h, 59
- bd\_open\_file\_dec
  - bluray.h, 59
- bd\_open\_files
  - bluray.h, 60
- bd\_open\_stream
  - bluray.h, 60
- BD\_OVERLAY, 15
- BD\_OVERLAY\_CLEAR
  - overlay.h, 83
- BD\_OVERLAY\_CLOSE
  - overlay.h, 83
- bd\_overlay\_cmd\_e
  - overlay.h, 82
- BD\_OVERLAY\_DRAW
  - overlay.h, 83
- BD\_OVERLAY\_FLUSH
  - overlay.h, 83
- BD\_OVERLAY\_HIDE
  - overlay.h, 83
- BD\_OVERLAY\_IG
  - overlay.h, 83
- BD\_OVERLAY\_INIT
  - overlay.h, 83
- BD\_OVERLAY\_PG
  - overlay.h, 83
- bd\_overlay\_plane\_e
  - overlay.h, 83
- bd\_overlay\_proc\_f
  - bluray.h, 44
- BD\_OVERLAY\_WIPE
  - overlay.h, 83
- BD\_PG\_PALETTE\_ENTRY, 16
  - T, 16
- BD\_PG\_RLE\_ELEM, 17
- bd\_play
  - bluray.h, 61
- bd\_play\_title

- bluray.h, 61
- bd\_player\_setting
  - bluray.h, 46
- bd\_read
  - bluray.h, 62
- bd\_read\_ext
  - bluray.h, 62
- bd\_read\_file
  - bluray.h, 63
- bd\_read\_skip\_still
  - bluray.h, 63
- bd\_refcnt\_inc
  - overlay.h, 83
- bd\_register\_argb\_overlay\_proc
  - bluray.h, 63
- bd\_register\_dir
  - filesystem.h, 31
- bd\_register\_file
  - filesystem.h, 31
- bd\_register\_overlay\_proc
  - bluray.h, 64
- bd\_seamless\_angle\_change
  - bluray.h, 65
- bd\_seek
  - bluray.h, 65
- bd\_seek\_chapter
  - bluray.h, 65
- bd\_seek\_mark
  - bluray.h, 67
- bd\_seek\_playitem
  - bluray.h, 67
- bd\_seek\_time
  - bluray.h, 68
- bd\_select\_angle
  - bluray.h, 68
- bd\_select\_playlist
  - bluray.h, 68
- bd\_select\_stream
  - bluray.h, 69
- bd\_select\_title
  - bluray.h, 70
- bd\_set\_debug\_handler
  - log\_control.h, 97
- bd\_set\_debug\_mask
  - log\_control.h, 98
- bd\_set\_player\_setting
  - bluray.h, 70
- bd\_set\_player\_setting\_str
  - bluray.h, 70
- bd\_set\_rate
  - bluray.h, 71
- bd\_set\_scr
  - bluray.h, 71
- bd\_still\_mode\_e
  - bluray.h, 47
- bd\_tell
  - bluray.h, 72
- bd\_tell\_time
  - bluray.h, 72
- bd\_user\_input
  - bluray.h, 72
- bd\_video\_format\_e
  - bluray.h, 47
- bd\_video\_rate\_e
  - bluray.h, 48
- BD\_VK\_0
  - keys.h, 87
- BD\_VK\_1
  - keys.h, 87
- BD\_VK\_2
  - keys.h, 87
- BD\_VK\_3
  - keys.h, 87
- BD\_VK\_4
  - keys.h, 87
- BD\_VK\_5
  - keys.h, 87
- BD\_VK\_6
  - keys.h, 87
- BD\_VK\_7
  - keys.h, 87
- BD\_VK\_8
  - keys.h, 87
- BD\_VK\_9
  - keys.h, 87
- BD\_VK\_BLUE
  - keys.h, 87
- BD\_VK\_DOWN
  - keys.h, 87
- BD\_VK\_ENTER
  - keys.h, 87
- BD\_VK\_GREEN
  - keys.h, 87
- bd\_vk\_key\_e
  - keys.h, 87
- BD\_VK\_LEFT
  - keys.h, 87
- BD\_VK\_MOUSE\_ACTIVATE
  - keys.h, 87
- BD\_VK\_NONE
  - keys.h, 87
- BD\_VK\_POPUP
  - keys.h, 87
- BD\_VK\_RED
  - keys.h, 87
- BD\_VK\_RIGHT
  - keys.h, 87
- BD\_VK\_ROOT\_MENU

- keys.h, 87
- BD\_VK\_UP
  - keys.h, 87
- BD\_VL\_YELLOW
  - keys.h, 87
- bdj
  - BLURAY\_TITLE, 23
- bdj\_detected
  - BLURAY\_DISC\_INFO, 20
- BLURAY
  - bluray.h, 44
- bluray-version.h, 34, 35
  - BLURAY\_VERSION\_CODE, 34
- bluray.h, 35, 73
  - bd\_argb\_overlay\_proc\_f, 44
  - bd\_chapter\_pos, 48
  - bd\_close, 49
  - BD\_EVENT\_ANGLE, 45
  - BD\_EVENT\_AUDIO\_STREAM, 45
  - BD\_EVENT\_CHAPTER, 45
  - BD\_EVENT\_DISCONTINUITY, 45
  - bd\_event\_e, 45
  - BD\_EVENT\_ENCRYPTED, 45
  - BD\_EVENT\_END\_OF\_TITLE, 45
  - BD\_EVENT\_ERROR, 45
  - BD\_EVENT\_IDLE, 46
  - BD\_EVENT\_IG\_STREAM, 45
  - BD\_EVENT\_KEY\_INTEREST\_TABLE, 46
  - BD\_EVENT\_MENU, 46
  - bd\_event\_name, 49
  - BD\_EVENT\_NONE, 45
  - BD\_EVENT\_PG\_TEXTST, 45
  - BD\_EVENT\_PG\_TEXTST\_STREAM, 45
  - BD\_EVENT\_PIP\_PG\_TEXTST, 45
  - BD\_EVENT\_PIP\_PG\_TEXTST\_STREAM, 45
  - BD\_EVENT\_PLAYITEM, 45
  - BD\_EVENT\_PLAYLIST, 45
  - BD\_EVENT\_PLAYLIST\_STOP, 45
  - BD\_EVENT\_PLAYMARK, 45
  - BD\_EVENT\_POPUP, 46
  - BD\_EVENT\_READ\_ERROR, 45
  - BD\_EVENT\_SECONDARY\_AUDIO, 45
  - BD\_EVENT\_SECONDARY\_AUDIO\_STREAM, 45
  - BD\_EVENT\_SECONDARY\_VIDEO, 45
  - BD\_EVENT\_SECONDARY\_VIDEO\_SIZE, 45
  - BD\_EVENT\_SECONDARY\_VIDEO\_STREAM, 45
  - BD\_EVENT\_SEEK, 45
  - BD\_EVENT\_SOUND\_EFFECT, 45
  - BD\_EVENT\_STEREOSCOPIC\_STATUS, 46
  - BD\_EVENT\_STILL, 45
  - BD\_EVENT\_STILL\_TIME, 45
  - BD\_EVENT\_TITLE, 45
  - BD\_EVENT\_UO\_MASK\_CHANGED, 46
  - bd\_free\_clpi, 49
  - bd\_free\_title\_info, 50
  - bd\_get\_clpi, 50
  - bd\_get\_current\_angle, 50
  - bd\_get\_current\_chapter, 51
  - bd\_get\_current\_title, 51
  - bd\_get\_disc\_info, 51
  - bd\_get\_event, 52
  - bd\_get\_main\_title, 52
  - bd\_get\_meta, 53
  - bd\_get\_meta\_file, 53
  - bd\_get\_playlist\_info, 54
  - bd\_get\_sound\_effect, 54
  - bd\_get\_title\_info, 55
  - bd\_get\_title\_size, 55
  - bd\_get\_titles, 56
  - bd\_get\_version, 56
  - bd\_init, 57
  - bd\_mark\_type\_e, 46
  - bd\_menu\_call, 57
  - bd\_mouse\_select, 57
  - bd\_open, 58
  - bd\_open\_dir, 58
  - bd\_open\_disc, 59
  - bd\_open\_file\_dec, 59
  - bd\_open\_files, 60
  - bd\_open\_stream, 60
  - bd\_overlay\_proc\_f, 44
  - bd\_play, 61
  - bd\_play\_title, 61
  - bd\_player\_setting, 46
  - bd\_read, 62
  - bd\_read\_ext, 62
  - bd\_read\_file, 63
  - bd\_read\_skip\_still, 63
  - bd\_register\_argb\_overlay\_proc, 63
  - bd\_register\_overlay\_proc, 64
  - bd\_seamless\_angle\_change, 65
  - bd\_seek, 65
  - bd\_seek\_chapter, 65
  - bd\_seek\_mark, 67
  - bd\_seek\_playitem, 67
  - bd\_seek\_time, 68
  - bd\_select\_angle, 68
  - bd\_select\_playlist, 68
  - bd\_select\_stream, 69
  - bd\_select\_title, 70
  - bd\_set\_player\_setting, 70
  - bd\_set\_player\_setting\_str, 70
  - bd\_set\_rate, 71
  - bd\_set\_scr, 71
  - bd\_still\_mode\_e, 47
  - bd\_tell, 72
  - bd\_tell\_time, 72
  - bd\_user\_input, 72



- bd\_video\_format\_e, 47
- bd\_video\_rate\_e, 48
- BLURAY, 44
- BLURAY\_MARK\_ENTRY, 46
- BLURAY\_MARK\_LINK, 46
- BLURAY\_PLAYER\_CACHE\_ROOT, 47
- BLURAY\_PLAYER\_JAVA\_HOME, 47
- BLURAY\_PLAYER\_PERSISTENT\_ROOT, 47
- BLURAY\_PLAYER\_SETTING\_3D\_CAP, 47
- BLURAY\_PLAYER\_SETTING\_AUDIO\_CAP, 46
- BLURAY\_PLAYER\_SETTING\_AUDIO\_LANG, 46
- BLURAY\_PLAYER\_SETTING\_COUNTRY\_CODE, 46
- BLURAY\_PLAYER\_SETTING\_DECODE\_PG, 47
- BLURAY\_PLAYER\_SETTING\_DISPLAY\_CAP, 46
- BLURAY\_PLAYER\_SETTING\_HDR\_PREFERENCE, 47
- BLURAY\_PLAYER\_SETTING\_MENU\_LANG, 46
- BLURAY\_PLAYER\_SETTING\_OUTPUT\_PREFER, 46
- BLURAY\_PLAYER\_SETTING\_PARENTAL, 46
- BLURAY\_PLAYER\_SETTING\_PERSISTENT\_STORAGE, 47
- BLURAY\_PLAYER\_SETTING\_PG\_LANG, 46
- BLURAY\_PLAYER\_SETTING\_PLAYER\_PROFILE, 47
- BLURAY\_PLAYER\_SETTING\_REGION\_CODE, 46
- BLURAY\_PLAYER\_SETTING\_SDR\_CONV\_PREFER, 47
- BLURAY\_PLAYER\_SETTING\_TEXT\_CAP, 47
- BLURAY\_PLAYER\_SETTING\_UHD\_CAP, 47
- BLURAY\_PLAYER\_SETTING\_UHD\_DISPLAY\_CAP, 47
- BLURAY\_PLAYER\_SETTING\_VIDEO\_CAP, 46
- BLURAY\_STILL\_INFINITE, 47
- BLURAY\_STILL\_NONE, 47
- BLURAY\_STILL\_TIME, 47
- BLURAY\_VIDEO\_FORMAT\_1080I, 48
- BLURAY\_VIDEO\_FORMAT\_1080P, 48
- BLURAY\_VIDEO\_FORMAT\_2160P, 48
- BLURAY\_VIDEO\_FORMAT\_480I, 48
- BLURAY\_VIDEO\_FORMAT\_480P, 48
- BLURAY\_VIDEO\_FORMAT\_576I, 48
- BLURAY\_VIDEO\_FORMAT\_576P, 48
- BLURAY\_VIDEO\_FORMAT\_720P, 48
- BLURAY\_VIDEO\_RATE\_24, 48
- BLURAY\_VIDEO\_RATE\_24000\_1001, 48
- BLURAY\_VIDEO\_RATE\_25, 48
- BLURAY\_VIDEO\_RATE\_30000\_1001, 48
- BLURAY\_VIDEO\_RATE\_50, 48
- BLURAY\_VIDEO\_RATE\_60000\_1001, 48
- BLURAY\_ACAP\_DD\_STEREO\_ONLY
  - player\_settings.h, 91
- BLURAY\_ACAP\_DD\_SURROUND
  - player\_settings.h, 91
- BLURAY\_ACAP\_DDPLUS\_DEP\_NONE
  - player\_settings.h, 91
- BLURAY\_ACAP\_DDPLUS\_DEP\_STEREO\_ONLY
  - player\_settings.h, 91
- BLURAY\_ACAP\_DDPLUS\_DEP\_SURROUND
  - player\_settings.h, 91
- BLURAY\_ACAP\_DDPLUS\_STEREO\_ONLY
  - player\_settings.h, 90
- BLURAY\_ACAP\_DDPLUS\_SURROUND
  - player\_settings.h, 91
- BLURAY\_ACAP\_DTSHD\_CORE\_STEREO\_ONLY
  - player\_settings.h, 91
- BLURAY\_ACAP\_DTSHD\_CORE\_SURROUND
  - player\_settings.h, 91
- BLURAY\_ACAP\_DTSHD\_EXT\_NONE
  - player\_settings.h, 91
- BLURAY\_ACAP\_DTSHD\_EXT\_STEREO\_ONLY
  - player\_settings.h, 91
- BLURAY\_ACAP\_DTSHD\_EXT\_SURROUND
  - player\_settings.h, 91
- BLURAY\_ACAP\_LPCM\_192\_NONE
  - player\_settings.h, 90
- BLURAY\_ACAP\_LPCM\_192\_STEREO\_ONLY
  - player\_settings.h, 90
- BLURAY\_ACAP\_LPCM\_192\_SURROUND
  - player\_settings.h, 90
- BLURAY\_ACAP\_LPCM\_48\_96\_STEREO\_ONLY
  - player\_settings.h, 90
- BLURAY\_ACAP\_LPCM\_48\_96\_SURROUND
  - player\_settings.h, 90
- BLURAY\_ACAP\_MLP\_NONE
  - player\_settings.h, 91
- BLURAY\_ACAP\_MLP\_STEREO\_ONLY
  - player\_settings.h, 91
- BLURAY\_ACAP\_MLP\_SURROUND
  - player\_settings.h, 91
- BLURAY\_CLIP\_INFO, 17
- BLURAY\_DISC\_INFO, 18
  - bdj\_detected, 20
  - first\_play, 20
  - no\_menu\_support, 21
  - top\_menu, 21
- BLURAY\_MARK\_ENTRY
  - bluray.h, 46
- BLURAY\_MARK\_LINK
  - bluray.h, 46
- BLURAY\_OUTPUT\_PREFER\_2D
  - player\_settings.h, 92
- BLURAY\_OUTPUT\_PREFER\_3D
  - player\_settings.h, 92
- BLURAY\_PERSISTENT\_STORAGE\_DISABLE
  - player\_settings.h, 93
- BLURAY\_PERSISTENT\_STORAGE\_ENABLE

- player\_settings.h, [93](#)
- BLURAY\_PG\_TEXTST\_DECODER\_DISABLE
  - player\_settings.h, [93](#)
- BLURAY\_PG\_TEXTST\_DECODER\_ENABLE
  - player\_settings.h, [93](#)
- BLURAY\_PLAYER\_CACHE\_ROOT
  - bluray.h, [47](#)
- BLURAY\_PLAYER\_JAVA\_HOME
  - bluray.h, [47](#)
- BLURAY\_PLAYER\_PERSISTENT\_ROOT
  - bluray.h, [47](#)
- BLURAY\_PLAYER\_PROFILE\_1\_v1\_0
  - player\_settings.h, [92](#)
- BLURAY\_PLAYER\_PROFILE\_1\_v1\_1
  - player\_settings.h, [92](#)
- BLURAY\_PLAYER\_PROFILE\_2\_v2\_0
  - player\_settings.h, [92](#)
- BLURAY\_PLAYER\_PROFILE\_3\_v2\_0
  - player\_settings.h, [92](#)
- BLURAY\_PLAYER\_PROFILE\_5\_v2\_4
  - player\_settings.h, [93](#)
- BLURAY\_PLAYER\_PROFILE\_6\_v3\_0
  - player\_settings.h, [93](#)
- BLURAY\_PLAYER\_PROFILE\_6\_v3\_1
  - player\_settings.h, [93](#)
- BLURAY\_PLAYER\_SETTING\_3D\_CAP
  - bluray.h, [47](#)
- BLURAY\_PLAYER\_SETTING\_AUDIO\_CAP
  - bluray.h, [46](#)
- BLURAY\_PLAYER\_SETTING\_AUDIO\_LANG
  - bluray.h, [46](#)
- BLURAY\_PLAYER\_SETTING\_COUNTRY\_CODE
  - bluray.h, [46](#)
- BLURAY\_PLAYER\_SETTING\_DECODE\_PG
  - bluray.h, [47](#)
- BLURAY\_PLAYER\_SETTING\_DISPLAY\_CAP
  - bluray.h, [46](#)
- BLURAY\_PLAYER\_SETTING\_HDR\_PREFERENCE
  - bluray.h, [47](#)
- BLURAY\_PLAYER\_SETTING\_MENU\_LANG
  - bluray.h, [46](#)
- BLURAY\_PLAYER\_SETTING\_OUTPUT\_PREFER
  - bluray.h, [46](#)
- BLURAY\_PLAYER\_SETTING\_PARENTAL
  - bluray.h, [46](#)
- BLURAY\_PLAYER\_SETTING\_PERSISTENT\_STORAGE
  - bluray.h, [47](#)
- BLURAY\_PLAYER\_SETTING\_PG\_LANG
  - bluray.h, [46](#)
- BLURAY\_PLAYER\_SETTING\_PLAYER\_PROFILE
  - bluray.h, [47](#)
- BLURAY\_PLAYER\_SETTING\_REGION\_CODE
  - bluray.h, [46](#)
- BLURAY\_PLAYER\_SETTING\_SDR\_CONV\_PREFER
  - bluray.h, [47](#)
- bluray.h, [47](#)
- BLURAY\_PLAYER\_SETTING\_TEXT\_CAP
  - bluray.h, [47](#)
- BLURAY\_PLAYER\_SETTING\_UHD\_CAP
  - bluray.h, [47](#)
- BLURAY\_PLAYER\_SETTING\_UHD\_DISPLAY\_CAP
  - bluray.h, [47](#)
- BLURAY\_PLAYER\_SETTING\_VIDEO\_CAP
  - bluray.h, [46](#)
- BLURAY\_REGION\_A
  - player\_settings.h, [91](#)
- BLURAY\_REGION\_B
  - player\_settings.h, [91](#)
- BLURAY\_REGION\_C
  - player\_settings.h, [91](#)
- BLURAY\_SOUND\_EFFECT, [21](#)
  - samples, [22](#)
- BLURAY\_STILL\_INFINITE
  - bluray.h, [47](#)
- BLURAY\_STILL\_NONE
  - bluray.h, [47](#)
- BLURAY\_STILL\_TIME
  - bluray.h, [47](#)
- BLURAY\_STREAM\_INFO, [22](#)
- BLURAY\_TITLE, [23](#)
  - bdj, [23](#)
- BLURAY\_TITLE\_CHAPTER, [24](#)
- BLURAY\_TITLE\_INFO, [24](#)
- BLURAY\_TITLE\_MARK, [25](#)
- BLURAY\_VCAP\_25Hz\_50Hz
  - player\_settings.h, [92](#)
- BLURAY\_VCAP\_SECONDARY\_HD
  - player\_settings.h, [92](#)
- BLURAY\_VERSION\_CODE
  - bluray-version.h, [34](#)
- BLURAY\_VIDEO\_FORMAT\_1080I
  - bluray.h, [48](#)
- BLURAY\_VIDEO\_FORMAT\_1080P
  - bluray.h, [48](#)
- BLURAY\_VIDEO\_FORMAT\_2160P
  - bluray.h, [48](#)
- BLURAY\_VIDEO\_FORMAT\_480I
  - bluray.h, [48](#)
- BLURAY\_VIDEO\_FORMAT\_480P
  - bluray.h, [48](#)
- BLURAY\_VIDEO\_FORMAT\_576I
  - bluray.h, [48](#)
- BLURAY\_VIDEO\_FORMAT\_576P
  - bluray.h, [48](#)
- BLURAY\_VIDEO\_FORMAT\_720P
  - bluray.h, [48](#)
- BLURAY\_VIDEO\_RATE\_24
  - bluray.h, [48](#)
- BLURAY\_VIDEO\_RATE\_24000\_1001

- bluray.h, [48](#)
- BLURAY\_VIDEO\_RATE\_25
  - bluray.h, [48](#)
- BLURAY\_VIDEO\_RATE\_30000\_1001
  - bluray.h, [48](#)
- BLURAY\_VIDEO\_RATE\_50
  - bluray.h, [48](#)
- BLURAY\_VIDEO\_RATE\_60000\_1001
  - bluray.h, [48](#)
- buf
  - BD\_ARGB\_BUFFER, [8](#)
- close
  - BD\_DIR\_H, [10](#)
  - BD\_FILE\_H, [12](#)
- DBG\_BDJ
  - log\_control.h, [97](#)
- DBG\_BLURAY
  - log\_control.h, [97](#)
- DBG\_CRIT
  - log\_control.h, [97](#)
- DBG\_DECODE
  - log\_control.h, [97](#)
- DBG\_DIR
  - log\_control.h, [97](#)
- DBG\_GC
  - log\_control.h, [97](#)
- DBG\_HDMV
  - log\_control.h, [97](#)
- DBG\_JNI
  - log\_control.h, [97](#)
- DBG\_NAV
  - log\_control.h, [97](#)
- DBG\_STREAM
  - log\_control.h, [97](#)
- debug\_mask\_t
  - log\_control.h, [97](#)
- dirty
  - BD\_ARGB\_BUFFER, [8](#)
- eof
  - BD\_FILE\_H, [12](#)
- filesystem.h, [29](#), [32](#)
  - BD\_DIR\_OPEN, [30](#)
  - BD\_FILE\_OPEN, [30](#)
  - bd\_register\_dir, [31](#)
  - bd\_register\_file, [31](#)
- first\_play
  - BLURAY\_DISC\_INFO, [20](#)
- internal
  - BD\_FILE\_H, [13](#)
- keys.h, [86](#), [87](#)
- BD\_VK\_0, [87](#)
- BD\_VK\_1, [87](#)
- BD\_VK\_2, [87](#)
- BD\_VK\_3, [87](#)
- BD\_VK\_4, [87](#)
- BD\_VK\_5, [87](#)
- BD\_VK\_6, [87](#)
- BD\_VK\_7, [87](#)
- BD\_VK\_8, [87](#)
- BD\_VK\_9, [87](#)
- BD\_VK\_BLUE, [87](#)
- BD\_VK\_DOWN, [87](#)
- BD\_VK\_ENTER, [87](#)
- BD\_VK\_GREEN, [87](#)
- bd\_vk\_key\_e, [87](#)
- BD\_VK\_LEFT, [87](#)
- BD\_VK\_MOUSE\_ACTIVATE, [87](#)
- BD\_VK\_NONE, [87](#)
- BD\_VK\_POPUP, [87](#)
- BD\_VK\_RED, [87](#)
- BD\_VK\_RIGHT, [87](#)
- BD\_VK\_ROOT\_MENU, [87](#)
- BD\_VK\_UP, [87](#)
- BD\_VL\_YELLOW, [87](#)
- log\_control.h, [95](#), [98](#)
  - bd\_get\_debug\_mask, [97](#)
  - BD\_LOG\_FUNC, [96](#)
  - bd\_set\_debug\_handler, [97](#)
  - bd\_set\_debug\_mask, [98](#)
  - DBG\_BDJ, [97](#)
  - DBG\_BLURAY, [97](#)
  - DBG\_CRIT, [97](#)
  - DBG\_DECODE, [97](#)
  - DBG\_DIR, [97](#)
  - DBG\_GC, [97](#)
  - DBG\_HDMV, [97](#)
  - DBG\_JNI, [97](#)
  - DBG\_NAV, [97](#)
  - DBG\_STREAM, [97](#)
  - debug\_mask\_t, [97](#)
- meta\_data.h, [33](#)
- META\_DL, [26](#)
- META\_THUMBNAIL, [27](#)
- META\_TITLE, [27](#)
- no\_menu\_support
  - BLURAY\_DISC\_INFO, [21](#)
- overlay.h, [81](#), [83](#)
  - BD\_ARGB\_OVERLAY\_CLOSE, [82](#)
  - bd\_argb\_overlay\_cmd\_e, [82](#)
  - BD\_ARGB\_OVERLAY\_DRAW, [82](#)
  - BD\_ARGB\_OVERLAY\_FLUSH, [82](#)

- BD\_ARGB\_OVERLAY\_INIT, [82](#)
- BD\_OVERLAY\_CLEAR, [83](#)
- BD\_OVERLAY\_CLOSE, [83](#)
- bd\_overlay\_cmd\_e, [82](#)
- BD\_OVERLAY\_DRAW, [83](#)
- BD\_OVERLAY\_FLUSH, [83](#)
- BD\_OVERLAY\_HIDE, [83](#)
- BD\_OVERLAY\_IG, [83](#)
- BD\_OVERLAY\_INIT, [83](#)
- BD\_OVERLAY\_PG, [83](#)
- bd\_overlay\_plane\_e, [83](#)
- BD\_OVERLAY\_WIPE, [83](#)
- bd\_refcnt\_inc, [83](#)
- player\_settings.h, [88](#), [94](#)
  - BLURAY\_ACAP\_DD\_STEREO\_ONLY, [91](#)
  - BLURAY\_ACAP\_DD\_SURROUND, [91](#)
  - BLURAY\_ACAP\_DDPLUS\_DEP\_NONE, [91](#)
  - BLURAY\_ACAP\_DDPLUS\_DEP\_STEREO\_ONLY, [91](#)
  - BLURAY\_ACAP\_DDPLUS\_DEP\_SURROUND, [91](#)
  - BLURAY\_ACAP\_DDPLUS\_STEREO\_ONLY, [90](#)
  - BLURAY\_ACAP\_DDPLUS\_SURROUND, [91](#)
  - BLURAY\_ACAP\_DTSHD\_CORE\_STEREO\_ONLY, [91](#)
  - BLURAY\_ACAP\_DTSHD\_CORE\_SURROUND, [91](#)
  - BLURAY\_ACAP\_DTSHD\_EXT\_NONE, [91](#)
  - BLURAY\_ACAP\_DTSHD\_EXT\_STEREO\_ONLY, [91](#)
  - BLURAY\_ACAP\_DTSHD\_EXT\_SURROUND, [91](#)
  - BLURAY\_ACAP\_LPCM\_192\_NONE, [90](#)
  - BLURAY\_ACAP\_LPCM\_192\_STEREO\_ONLY, [90](#)
  - BLURAY\_ACAP\_LPCM\_192\_SURROUND, [90](#)
  - BLURAY\_ACAP\_LPCM\_48\_96\_STEREO\_ONLY, [90](#)
  - BLURAY\_ACAP\_LPCM\_48\_96\_SURROUND, [90](#)
  - BLURAY\_ACAP\_MLP\_NONE, [91](#)
  - BLURAY\_ACAP\_MLP\_STEREO\_ONLY, [91](#)
  - BLURAY\_ACAP\_MLP\_SURROUND, [91](#)
  - BLURAY\_OUTPUT\_PREFER\_2D, [92](#)
  - BLURAY\_OUTPUT\_PREFER\_3D, [92](#)
  - BLURAY\_PERSISTENT\_STORAGE\_DISABLE, [93](#)
  - BLURAY\_PERSISTENT\_STORAGE\_ENABLE, [93](#)
  - BLURAY\_PG\_TEXTST\_DECODER\_DISABLE, [93](#)
  - BLURAY\_PG\_TEXTST\_DECODER\_ENABLE, [93](#)
  - BLURAY\_PLAYER\_PROFILE\_1\_v1\_0, [92](#)
  - BLURAY\_PLAYER\_PROFILE\_1\_v1\_1, [92](#)
  - BLURAY\_PLAYER\_PROFILE\_2\_v2\_0, [92](#)
  - BLURAY\_PLAYER\_PROFILE\_3\_v2\_0, [92](#)
  - BLURAY\_PLAYER\_PROFILE\_5\_v2\_4, [93](#)
  - BLURAY\_PLAYER\_PROFILE\_6\_v3\_0, [93](#)
  - BLURAY\_PLAYER\_PROFILE\_6\_v3\_1, [93](#)
  - BLURAY\_REGION\_A, [91](#)
  - BLURAY\_REGION\_B, [91](#)
  - BLURAY\_REGION\_C, [91](#)
  - BLURAY\_VCAP\_25Hz\_50Hz, [92](#)
  - BLURAY\_VCAP\_SECONDARY\_HD, [92](#)
- read
  - BD\_DIR\_H, [10](#)
  - BD\_FILE\_H, [13](#)
- samples
  - BLURAY\_SOUND\_EFFECT, [22](#)
- seek
  - BD\_FILE\_H, [13](#)
- T
  - BD\_PG\_PALETTE\_ENTRY, [16](#)
- tell
  - BD\_FILE\_H, [14](#)
- top\_menu
  - BLURAY\_DISC\_INFO, [21](#)
- write
  - BD\_FILE\_H, [14](#)