Rocky Enterprise Linux 9.2 Manual Pages on command 'libopenjp2.3'

$ man libopenjp2.3

libopenjp2(3) Oct 2010 libopenjp2(3)

NAME

libopenjp2 - a library for reading and writing JPEG2000 image files.

SYNOPSIS

#include <openjpeg.h>

CONVERSION FORMATS

PGX: imagetopgx() / pgxtoimage()

PXM: imagetopnm() / pnmtoimage()

BMP: imagetobmp() / bmptoimage()

TIF: imagetotif() / tiftoimage()

RAW: imagetoraw() / rawtoimage()

TGA: imagetotga() / tgatoimage()

PNG: imagetopng() / pngtoimage()

YUV: imagetoyuv() / yuvtoimage() (MJ2)

READ

opj_set_default_decoder_parameters(opj_dparameters_t *params);

opj_dinfo_t *opj_create_decompress(OPJ_CODEC_FORMAT format);

opj_event_mgr_t *opj_set_event_mgr(opj_common_ptr info, opj_event_mgr_t

*event_mgr, void *context);

void opj_setup_decoder(opj_dinfo_t *dinfo, opj_dparameters_t * params);

opj_cio_t *opj_cio_open(opj_common_ptr info, unsigned char *buf, int

buf_len);

opj_image_t *opj_decode(opj_dinfo_t *dinfo, opj_cio_t *cio);

void opj_cio_close(opj_cio_t *cio);

void opj_destroy_decompress(opj_dinfo_t *dinfo);

void opj_image_destroy(opj_image_t *image);

WRITE

void opj_set_default_encoder_parameters(opj_cparameters_t *params);

/* opj_image_t *FORMATtoimage(const char *fname, opj_cparameters_t

*params);

*/

opj_cinfo_t* opj_create_compress(OPJ_CODEC_FORMAT format);

opj_event_mgr_t *opj_set_event_mgr(opj_common_ptr info, opj_event_mgr_t

*event_mgr, void *context);

void opj_setup_encoder(opj_cinfo_t *cinfo, opj_cparameters_t *params,

opj_image_t *image);

opj_cio_t *opj_cio_open(opj_common_ptr cinfo, NULL, 0);

bool opj_encode(opj_cinfo_t *cinfo, opj_cio_t *cio, opj_image_t *image,

char *index);

void opj_cio_close(opj_cio_t *cio);

void opj_destroy_compress(opj_cinfo_t *cinfo);

void opj_image_destroy(opj_image_t *image);

GENERAL

void opj_image_create(int numcmpts, opj_image_cmptparm_t *cmptparms,

OPJ_COLOR_SPACE clrspc);

int cio_tell(opj_cio_t *cio);

void cio_seek(opj_cio_t *cio, int pos);

opj_image_t *opj_decode_with_info(opj_dinfo_t *dinfo, opj_cio_t *cio,

opj_codestream_info_t *cstr_info);

bool opj_encode_with_info(opj_cinfo_t *cinfo, opj_cio_t *cio, opj_im?

age_t *image, opj_codestream_info_t *cstr_info);

void opj_destroy_cstr_info(opj_codestream_info_t *cstr_info);

const char *opj_version(void);

OPJ_CODEC_FORMAT

CODEC_J2K or CODEC_JPT or CODEC_JP2

OPJ_COLOR_SPACE

CLRSPC_UNKNOWN or CLRSPC_UNSPECIFIED or CLRSPC_SRGB or CLRSPC_GRAY or

CLRSPC_SYCC

DECOMPRESSION PARAMETERS

typedef struct opj_dparameters

{

/*

Set the number of highest resolution levels to be discarded.

The image resolution is effectively divided by 2 to the power

of the number of discarded levels.

The reduce factor is limited by the smallest total number of

decomposition levels among tiles.

if != 0, then original dimension divided by 2^(reduce);

if == 0 or not used, image is decoded to the full resolution

*/

int cp_reduce;

/*

Set the maximum number of quality layers to decode.

If there are less quality layers than the specified number,

all the quality layers are decoded.

if != 0, then only the first "layer" layers are decoded;

if == 0 or not used, all the quality layers are decoded

*/

int cp_layer;

/*command line encoder parameters (not used inside the library) */

/* input file name */

char infile[OPJ_PATH_LEN];

/* output file name */

char outfile[OPJ_PATH_LEN];

```
        /* input file format: see OPJ_CODEC_FORMAT */
        int decod_format;
        /* output file format */
        int cod_format;
        /*JPWL decoding parameters */
        /* activates the JPWL correction capabilities */
        bool jpwl_correct;
        /* expected number of components */
        int jpwl_exp_comps;
        /* maximum number of tiles */
        int jpwl_max_tiles;
        /*
        Specify whether the decoding should be done on the entire
        codestream, or be limited to the main header
        Limiting the decoding to the main header makes it possible
        to extract the characteristics of the codestream
        if == NO_LIMITATION, the entire codestream is decoded;
        if == LIMIT_TO_MAIN_HEADER, only the main header is decoded;
        */
        OPJ_LIMIT_DECODING cp_limit_decoding;
    } opj_dparameters_t;
```

COMPRESSION PARAMETERS

```
    typedef struct opj_cparameters
    {
        /* size of tile: tile_size_on = false (not in argument)
        or tile_size_on = true (in argument) */
        bool tile_size_on;
        /* XTOsiz */
        int cp_tx0;
        /* YTOsiz */
        int cp_ty0;
        /* XTsiz */
        int cp_tdx;
```

```
/* YTsiz */
int cp_tdy;
/* allocation by rate/distortion */
int cp_disto_alloc;
/* allocation by fixed layer */
int cp_fixed_alloc;
/* add fixed_quality */
int cp_fixed_quality;
/* fixed layer */
int *cp_matrice;
/* comment for coding */
char *cp_comment;
/* coding style */
int csty;
/* progression order:
   PROG_UNKNOWN, LRCP(default), RLCP, RPCL, PCRL, CPRL */
OPJ_PROG_ORDER prog_order;
/* progression order changes */
opj_poc_t POC[32];
/* number of progression order changes (POC), default: 0 */
int numpocs;
/* number of layers */
int tcp_numlayers;
/* rates of layers */
float tcp_rates[100];
/* different psnr for successive layers */
float tcp_distoratio[100];
/* number of resolutions */
int numresolution;
/* initial code block width, default: 64 */
int cblockw_init;
/* initial code block height, default: 64 */
int cblockh_init;
```

```
/* mode switch (cblk_style) */
/* 1 : use the irreversible DWT 9-7,
    0 : use lossless compression (default) */
int irreversible;
/* region of interest: affected component in [0..3],
    -1 means no ROI */
int roi_compno;
/* region of interest: upshift value */
int roi_shift;
/* number of precinct size specifications */
int res_spec;
/* initial precinct width */
int prcw_init[J2K_MAXRLVLS];
/* initial precinct height */
int prch_init[J2K_MAXRLVLS];
/*command line encoder parameters (not used inside the library) */
/* input file name */
char infile[OPJ_PATH_LEN];
/* output file name */
char outfile[OPJ_PATH_LEN];
/* DEPRECATED. Index generation is now handeld with the
    opj_encode_with_info() function. Set to NULL */
int index_on;
/* DEPRECATED. Index generation is now handeld with the
    opj_encode_with_info() function. Set to NULL */
char index[OPJ_PATH_LEN];
/* subimage encoding: origin image offset in x direction */
int image_offset_x0;
/* subimage encoding: origin image offset in y direction */
int image_offset_y0;
/* subsampling value for dx */
int subsampling_dx;
/* subsampling value for dy */
```

```
int subsampling_dy;
/* input file format */
int decod_format;
/* output file format: see OPJ_CODEC_FORMAT */
int cod_format;
/*JPWL encoding parameters */
/* enables writing of EPC in MH, thus activating JPWL */
bool jpwl_epc_on;
/* error protection method for MH (0,1,16,32,37-128) */
int jpwl_hprot_MH;
/* tile number of header protection specification (>=0) */
int jpwl_hprot_TPH_tileno[JPWL_MAX_NO_TILESPECS];
/* error protection methods for TPHs (0,1,16,32,37-128) */
int jpwl_hprot_TPH[JPWL_MAX_NO_TILESPECS];
/* tile number of packet protection specification (>=0) */
int jpwl_pprot_tileno[JPWL_MAX_NO_PACKSPECS];
/* packet number of packet protection specification (>=0) */
int jpwl_pprot_packno[JPWL_MAX_NO_PACKSPECS];
/* error protection methods for packets (0,1,16,32,37-128) */
int jpwl_pprot[JPWL_MAX_NO_PACKSPECS];
/* enables writing of ESD, (0=no/1/2 bytes) */
int jpwl_sens_size;
/* sensitivity addressing size (0=auto/2/4 bytes) */
int jpwl_sens_addr;
/* sensitivity range (0-3) */
int jpwl_sens_range;
/* sensitivity method for MH (-1=no,0-7) */
int jpwl_sens_MH;
/* tile number of sensitivity specification (>=0) */
int jpwl_sens_TPH_tileno[JPWL_MAX_NO_TILESPECS];
/* sensitivity methods for TPHs (-1=no,0-7) */
int jpwl_sens_TPH[JPWL_MAX_NO_TILESPECS];
/* Digital Cinema compliance: OFF-not compliant,
```

```
        CINEMA2K_24, CINEMA2K_48, CINEMA4K_24 */
    OPJ_CINEMA_MODE cp_cinema;
    /* Maximum rate for each component.
       If == 0, component size limitation is not considered */
    int max_comp_size;
    /* Profile name*/
    OPJ_RSIZ_CAPABILITIES cp_rsiz;
    /* Tile part generation*/
    char tp_on;
    /* Flag for Tile part generation*/
    char tp_flag;
    /* MCT (multiple component transform) */
    char tcp_mct;
} opj_cparameters_t;
```

AUTHORS

Copyright (c) 2002-2014, Universite catholique de Louvain (UCL), Bel?

gium

Copyright (c) 2002-2014, Professor Benoit Macq

Copyright (c) 2001-2003, David Janssens

Copyright (c) 2002-2003, Yannick Verschueren

Copyright (c) 2003-2007, Francois-Olivier Devaux and Antonin Descampe

Copyright (c) 2005, Herve Drolon, FreeImage Team

Copyright (c) 2006-2007, Parvatha Elangovan

SEE ALSO

image_to_j2k(1) j2k_to_image(1) j2k_dump(1)

JPWL_image_to_j2k(1) JPWL_j2k_to_image(1)

extract_j2k_from_mj2(1) wrap_j2k_in_mj2(1) frames_to_mj2(1)

mj2_to_frames(1)

Version 1.4.0 Oct 2010 libopenjp2(3)