

XviD API 2.1 Reference (for 0.9.x series)

Author: XviD Team

2003-07-28

Contents

1	XviD core library Module Index	1
1.1	XviD core library Modules	1
2	XviD core library Data Structure Index	3
2.1	XviD core library Data Structures	3
3	XviD core library Page Index	5
3.1	XviD core library Related Pages	5
4	XviD core library Module Documentation	7
4.1	Global constants used in both encoder and decoder.	7
4.2	API version	8
4.3	Error codes returned by XviD API entry points.	9
4.4	Colorspaces constants.	10
4.5	Initialization constants, structures and functions.	12
4.6	Flags for XVID_INIT_PARAM.cpu_flags.	13
4.7	x86 specific cpu flags	14
4.8	ia64 specific cpu flags.	15
4.9	Initialization commands.	16
4.10	Initialization entry point.	17
4.11	Decoder related functions and structures.	18
4.12	Flags for XVID_DEC_FRAME.general	19
4.13	Decoder operations	20
4.14	Decoder entry point	21
4.15	Encoder related functions and structures.	22
4.16	Flags for XVID_ENC_FRAME.general	23
4.17	Flags for XVID_ENC_FRAME.motion	27
4.18	Encoder operations	31
4.19	Encoder entry point	32

5	XviD core library Data Structure Documentation	33
5.1	XVID_ENC_FRAME Struct Reference	33
5.2	XVID_ENC_PARAM Struct Reference	36
5.3	XVID_ENC_STATS Struct Reference	39
5.4	XVID_INIT_PARAM Struct Reference	41
6	XviD core library Page Documentation	43
6.1	Todo List	43
6.2	Deprecated List	44

Chapter 1

XviD core library Module Index

1.1 XviD core library Modules

Here is a list of all modules:

Global constants used in both encoder and decoder.	7
API version	8
Error codes returned by XviD API entry points.	9
Colorspaces constants.	10
Initialization constants, structures and functions.	12
Flags for XVID_INIT_PARAM.cpu_flags.	13
x86 specific cpu flags	14
ia64 specific cpu flags.	15
Initialization commands.	16
Initialization entry point.	17
Decoder related functions and structures.	18
Flags for XVID_DEC_FRAME.general	19
Decoder operations	20
Decoder entry point	21
Encoder related functions and structures.	22
Flags for XVID_ENC_FRAME.general	23
Flags for XVID_ENC_FRAME.motion	27
Encoder operations	31
Encoder entry point	32

Chapter 2

XviD core library Data Structure Index

2.1 XviD core library Data Structures

Here are the data structures with brief descriptions:

XVID_ENC_FRAME (Structure used to pass a frame to the encoder)	33
XVID_ENC_PARAM (Structure used for encoder instance creation)	36
XVID_ENC_STATS (Encoding statistics)	39
XVID_INIT_PARAM (Structure used in xvid_init function)	41

Chapter 3

XviD core library Page Index

3.1 XviD core library Related Pages

Here is a list of all related documentation pages:

Todo List	43
Deprecated List	44

Chapter 4

XviD core library Module Documentation

4.1 Global constants used in both encoder and decoder.

4.1.1 Detailed Description

This module describe all constants used in both the encoder and the decoder.

Modules

- [API version](#)
- [Error codes returned by XviD API entry points.](#)
- [Colorspaces constants.](#)

4.2 API version

Defines

- `#define APL_VERSION ((2 << 16) | (1))`

This constant tells you what XviD's version this header defines.

4.2.1 Define Documentation

4.2.1.1 `#define APL_VERSION ((2 << 16) | (1))`

This constant tells you what XviD's version this header defines.

You can use it to check if the host XviD library API is the same as the one you used to build you client program. If versions mismatch, then it is highly possible that your application will segfault because the host XviD library and your application use different structures.

4.3 Error codes returned by XviD API entry points.

Defines

- `#define XVID_ERR_FAIL -1`
Operation failed.
- `#define XVID_ERR_OK 0`
Operation succeed.
- `#define XVID_ERR_MEMORY 1`
Operation failed.
- `#define XVID_ERR_FORMAT 2`
Operation failed.

4.3.1 Define Documentation

4.3.1.1 `#define XVID_ERR_FAIL -1`

Operation failed.

The requested XviD operation failed. If this error code is returned from :

- the `xvid_init` function : you must not try to use an XviD's instance from this point of the code. Clean all instances you already created and exit the program cleanly.
- `xvid_encore` or `xvid_decure` : something was wrong and en/decoding operation was not completed sucessfully. you can stop the en/decoding process or just ignore and go on.
- `xvid_stop` : you can safely ignore it if you call this function at the end of your program.

4.3.1.2 `#define XVID_ERR_FORMAT 2`

Operation failed.

The format of the parameters or input stream were incorrect.

4.3.1.3 `#define XVID_ERR_MEMORY 1`

Operation failed.

Insufficient memory was available on the host system.

4.3.1.4 `#define XVID_ERR_OK 0`

Operation succeed.

The requested XviD operation succeed, you can continue to use XviD's functions.

4.4 Colorspaces constants.

Defines

- #define [XVID_CSP_RGB24](#) 0
24-bit RGB colorspace (b,g,r packed)
- #define [XVID_CSP_YV12](#) 1
YV12 colorspace (y,v,u planar).
- #define [XVID_CSP_YUY2](#) 2
YUY2 colorspace (y,u,y,v packed).
- #define [XVID_CSP_UYVY](#) 3
UYVY colorspace (u,y,v,y packed).
- #define [XVID_CSP_I420](#) 4
I420 colorsapce (y,u,v planar).
- #define [XVID_CSP_RGB555](#) 10
16-bit RGB555 colorspace
- #define [XVID_CSP_RGB565](#) 11
16-bit RGB565 colorspace
- #define [XVID_CSP_USER](#) 12
user colorspace format, where the image buffer points to a DEC_PICTURE (y,u,v planar) structure
- #define [XVID_CSP_EXTERN](#) 1004
Special colorspace used for slice rendering.
- #define [XVID_CSP_YVYU](#) 1002
YVYU colorspace (y,v,y,u packed).
- #define [XVID_CSP_RGB32](#) 1000
32-bit RGB colorspace (b,g,r,a packed)
- #define [XVID_CSP_NULL](#) 9999
NULL colorspace; no conversion is performed.
- #define [XVID_CSP_VFLIP](#) 0x80000000
(flag) Flip frame vertically during conversion

4.4.1 Define Documentation

4.4.1.1 #define XVID_CSP_EXTERN 1004

Special colorspace used for slice rendering.

The application provides an external buffer to XviD. This way, XviD works directly into the final rendering buffer, no need to specify this is a speed boost feature. This feature is only used by mplayer at the moment, refer to mplayer code to see how it can be used.

4.4.1.2 `#define XVID_CSP_USER 12`

user colorspace format, where the image buffer points to a DEC_PICTURE (y,u,v planar) structure

For encoding, image is read from the DEC_PICTURE parameter values. For decoding, the DEC_PICTURE parameters are set, pointing to the internal XviD image buffer.

4.5 Initialization constants, structures and functions.

4.5.1 Detailed Description

This section describes all the constants, structures and functions used to initialize the XviD core library.

Modules

- [Flags for XVID_INIT_PARAM.cpu_flags.](#)

This section describes all constants that show host cpu available features, and allow a client application to force usage of some cpu instructions sets.

- [Initialization entry point.](#)

Data Structures

- struct [XVID_INIT_PARAM](#)

Structure used in xvid_init function.

4.6 Flags for XVID_INIT_PARAM.cpu_flags.

4.6.1 Detailed Description

This section describes all constants that show host cpu available features, and allow a client application to force usage of some cpu instructions sets.

Modules

- [x86 specific cpu flags](#)
- [ia64 specific cpu flags.](#)
- [Initialization commands.](#)

4.7 x86 specific cpu flags

Defines

- #define [XVID_CPU_MMX](#) 0x00000001
use/has MMX instruction set
- #define [XVID_CPU_MMXEXT](#) 0x00000002
use/has MMX-ext (pentium3) instruction set
- #define [XVID_CPU_SSE](#) 0x00000004
use/has SSE (pentium3) instruction set
- #define [XVID_CPU_SSE2](#) 0x00000008
use/has SSE2 (pentium4) instruction set
- #define [XVID_CPU_3DNOW](#) 0x00000010
use/has 3dNOW (k6-2) instruction set
- #define [XVID_CPU_3DNOWEXT](#) 0x00000020
use/has 3dNOW-ext (athlon) instruction set
- #define [XVID_CPU_TSC](#) 0x00000040
has TimeStampCounter instruction

4.8 ia64 specific cpu flags.

Defines

- `#define XVID_CPU_IA64 0x00000080`
Forces ia64 optimized code usage.

4.8.1 Define Documentation

4.8.1.1 `#define XVID_CPU_IA64 0x00000080`

Forces ia64 optimized code usage.

This flags allow client applications to force IA64 optimized functions. This feature is considered experimental and should be treated as is.

4.9 Initialization commands.

Defines

- `#define XVID_CPU_CHKONLY 0x40000000`
Check cpu features.
- `#define XVID_CPU_FORCE 0x80000000`
Force input flags to be used.

4.9.1 Define Documentation

4.9.1.1 `#define XVID_CPU_CHKONLY 0x40000000`

Check cpu features.

When this flag is set, the `xvid_init` function performs just a cpu feature checking and then fills the `cpu` field. This flag is usefull when client applications want to know what instruction sets the host cpu supports.

4.9.1.2 `#define XVID_CPU_FORCE 0x80000000`

Force input flags to be used.

When this flag is set, client application forces XviD to use other flags set in `cpu_flags`. **Use** this at your own risk.

4.10 Initialization entry point.

Functions

- `int xvid_init (void *handle, int opt, void *param1, void *param2)`
Initialization entry point.

4.10.1 Function Documentation

4.10.1.1 `int xvid_init (void * handle, int opt, void * param1, void * param2)`

Initialization entry point.

This is the XviD's initialization entry point, it is only used to initialize the XviD internal data (function pointers, vector length code tables, rgb2yuv lookup tables).

Parameters:

- handle* Reserved for future use.
- opt* Reserved for future use (set it to 0).
- param1* Used to pass an `XVID_INIT_PARAM` parameter.
- param2* Reserved for future use.

4.11 Decoder related functions and structures.

4.11.1 Detailed Description

This part describes all the structures/functions from XviD's API needed for decoding a MPEG4 compliant streams.

Modules

- [Flags for XVID_DEC_FRAME.general](#)
Flags' description for the XVID_DEC_FRAME.general member.
- [Decoder operations](#)
These are all the operations XviD's decoder can perform.
- [Decoder entry point](#)

Data Structures

- struct **XVID_DEC_FRAME**
- struct **XVID_DEC_PARAM**
- struct **XVID_DEC_PICTURE**

4.12 Flags for XVID_DEC_FRAME.general

4.12.1 Detailed Description

Flags' description for the XVID_DEC_FRAME.general member.

Defines

- #define [XVID_QUICK_DECODE](#) 0x00000010
Not used at the moment.

4.13 Decoder operations

4.13.1 Detailed Description

These are all the operations XviD's decoder can perform.

Defines

- #define `XVID_DEC_DECODE` 0
Decodes a frame.
- #define `XVID_DEC_CREATE` 1
Creates a decoder instance.
- #define `XVID_DEC_DESTROY` 2
Destroys a decoder instance.

4.13.2 Define Documentation

4.13.2.1 #define `XVID_DEC_CREATE` 1

Creates a decoder instance.

This operation constant is used by a client application in order to create a decoder instance. Decoder instances are independant from each other, and can be safely threaded.

4.13.2.2 #define `XVID_DEC_DECODE` 0

Decodes a frame.

This operation constant is used when client application wants to decode a frame. Client application must also fill `XVID_DEC_FRAME` appropriately.

4.13.2.3 #define `XVID_DEC_DESTROY` 2

Destroys a decoder instance.

This operation constant is used by the client application to destroy a previously created decoder instance.

4.14 Decoder entry point

Functions

- int [xvid_decore](#) (void *handle, int opt, void *param1, void *param2)
Decoder entry point.

4.14.1 Function Documentation

4.14.1.1 int xvid_decore (void * *handle*, int *opt*, void * *param1*, void * *param2*)

Decoder entry point.

This is the XviD's decoder entry point. The possible operations are described in the [Decoder operations](#) section.

Parameters:

handle Decoder instance handle.

opt Decoder option constant

param1 Used to pass a XVID_DEC_PARAM or XVID_DEC_FRAME structure

param2 Reserved for future use.

4.15 Encoder related functions and structures.

Modules

- [Flags for XVID_ENC_FRAME.general](#)
- [Flags for XVID_ENC_FRAME.motion](#)
- [Encoder operations](#)

These are all the operations XviD's encoder can perform.

- [Encoder entry point](#)

Data Structures

- struct **HINTINFO**
- struct **MVBLOCKHINT**
- struct **MVFRAMEHINT**
- struct **VECTOR**
- struct [XVID_ENC_FRAME](#)

Structure used to pass a frame to the encoder.

- struct [XVID_ENC_PARAM](#)

Structure used for encoder instance creation.

- struct [XVID_ENC_STATS](#)

Encoding statistics.

4.16 Flags for XVID_ENC_FRAME.general

Defines

- #define [XVID_VALID_FLAGS](#) 0x80000000
Reserved for future use.
- #define [XVID_CUSTOM_QMATRIX](#) 0x00000004
Use custom quantization matrices.
- #define [XVID_H263QUANT](#) 0x00000010
Use H263 quantization.
- #define [XVID_MPEGQUANT](#) 0x00000020
Use MPEG4 quantization.
- #define [XVID_HALFPEL](#) 0x00000040
Halfpel motion estimation.
- #define [XVID_ADAPTIVEQUANT](#) 0x00000080
Adaptive quantization.
- #define [XVID_LUMIMASKING](#) 0x00000100
Lumimasking flag.
- #define [XVID_LATEINTRA](#) 0x00000200
Unknown.
- #define [XVID_INTERLACING](#) 0x00000400
MPEG4 interlacing mode.
- #define [XVID_TOPFIELDFIRST](#) 0x00000800
Unknown.
- #define [XVID_ALTERNATESCAN](#) 0x00001000
- #define [XVID_HINTEDME_GET](#) 0x00002000
Gets Motion vector data from ME system.
- #define [XVID_HINTEDME_SET](#) 0x00004000
Gives Motion vectors hint to ME system.
- #define [XVID_INTER4V](#) 0x00008000
Inter4V mode.
- #define [XVID_ME_ZERO](#) 0x00010000
Unused.
- #define [XVID_ME_LOGARITHMIC](#) 0x00020000
Unused.

- `#define XVID_ME_FULLSEARCH 0x00040000`
Unused.
- `#define XVID_ME_PMVFAST 0x00080000`
Use PMVfast ME algorithm.
- `#define XVID_ME_EPZS 0x00100000`
Use EPZS ME algorithm.
- `#define XVID_GREYSCALE 0x01000000`
Discard chroma data.
- `#define XVID_GRAYSCALE XVID_GREYSCALE`
XVID_GREYSCALE alias.

4.16.1 Define Documentation

4.16.1.1 `#define XVID_ADAPTIVEQUANT 0x00000080`

Adaptive quantization.

informs xvid to perform an adaptive quantization using a Luminance masking algorithm

4.16.1.2 `#define XVID_ALTERNATESCAN 0x00001000`

Deprecated

This flag is no longer used.

4.16.1.3 `#define XVID_CUSTOM_QMATRIX 0x00000004`

Use custom quantization matrices.

This flag forces XviD to use custom matrices passed to encoder in `XVID_ENC_FRAME` structure (members `quant_intra_matrix` and `quant_inter_matrix`)

4.16.1.4 `#define XVID_GRAYSCALE XVID_GREYSCALE`

XVID_GREYSCALE alias.

United States locale support.

4.16.1.5 `#define XVID_GREYSCALE 0x01000000`

Discard chroma data.

This flag forces XviD to discard chroma data, this is not mpeg4 greyscale mode, it simply drops chroma MBs using `cbp == 0` for these blocks

4.16.1.6 #define XVID_H263QUANT 0x00000010

Use H263 quantization.

This flag forces XviD to use H263 quantization type

4.16.1.7 #define XVID_HALFPEL 0x00000040

Halfpel motion estimation.

informs xvid to perform a half pixel motion estimation.

4.16.1.8 #define XVID_HINTEDME_GET 0x00002000

Gets Motion vector data from ME system.

informs xvid to return Motion Estimation vectors from the ME encoder algorithm. Used during a first pass.

4.16.1.9 #define XVID_HINTEDME_SET 0x00004000

Gives Motion vectors hint to ME system.

informs xvid to use the user given motion estimation vectors as hints for the encoder ME algorithms. Used during a 2nd pass.

4.16.1.10 #define XVID_INTER4V 0x00008000

Inter4V mode.

forces XviD to search a vector for each 8x8 block within the 16x16 Macro Block. This mode should be used only if the XVID_HALFPEL mode is activated (this could change in the future).

4.16.1.11 #define XVID_INTERLACING 0x00000400

MPEG4 interlacing mode.

Enables interlacing encoding mode

4.16.1.12 #define XVID_LATEINTRA 0x00000200

Unknown.

Deprecated

This flag is no longer used.

4.16.1.13 #define XVID_LUMIMASKING 0x00000100

Lumimasking flag.

Deprecated

This flag is no longer used.

4.16.1.14 #define XVID_ME_EPZS 0x00100000

Use EPZS ME algorithm.

Switches XviD ME algorithm to EPZS

4.16.1.15 #define XVID_ME_FULLSEARCH 0x00040000

Unused.

Do not use this flag (reserved for future use)

4.16.1.16 #define XVID_ME_LOGARITHMIC 0x00020000

Unused.

Do not use this flag (reserved for future use)

4.16.1.17 #define XVID_ME_PMVFAST 0x00080000

Use PMVfast ME algorithm.

Switches XviD ME algorithm to PMVfast

4.16.1.18 #define XVID_ME_ZERO 0x00010000

Unused.

Do not use this flag (reserved for future use)

4.16.1.19 #define XVID_MPEGQUANT 0x00000020

Use MPEG4 quantization.

This flag forces XviD to use MPEG4 quantization type

4.16.1.20 #define XVID_TOPFIELDFIRST 0x00000800

Unknown.

Deprecated

This flag is no longer used.

4.17 Flags for XVID_ENC_FRAME.motion

Defines

- #define `PMV_ADVANCEDDIAMOND8` 0x00004000
Uses advanced diamonds for 8x8 blocks.
- #define `PMV_ADVANCEDDIAMOND16` 0x00008000
Uses advanced diamonds for 16x16 blocks.
- #define `PMV_HALFPELDIAMOND16` 0x00010000
Turns on halfpel precision for 16x16 blocks.
- #define `PMV_HALFPELREFINE16` 0x00020000
Turns on halfpel refinement step.
- #define `PMV_EXTSEARCH16` 0x00040000
Extends search for 16x16 blocks.
- #define `PMV_EARLYSTOP16` 0x00080000
Dynamic ME thresholding.
- #define `PMV_QUICKSTOP16` 0x00100000
Dynamic ME thresholding.
- #define `PMV_UNRESTRICTED16` 0x00200000
Not implemented.
- #define `PMV_OVERLAPPING16` 0x00400000
Not implemented.
- #define `PMV_USESQUARES16` 0x00800000
Use square pattern.
- #define `PMV_HALFPELDIAMOND8` 0x01000000
see 16x16 equivalent
- #define `PMV_HALFPELREFINE8` 0x02000000
see 16x16 equivalent
- #define `PMV_EXTSEARCH8` 0x04000000
see 16x16 equivalent
- #define `PMV_EARLYSTOP8` 0x08000000
see 16x16 equivalent
- #define `PMV_QUICKSTOP8` 0x10000000
see 16x16 equivalent
- #define `PMV_UNRESTRICTED8` 0x20000000

see 16x16 equivalent

- `#define PMV_OVERLAPPING8 0x40000000`

see 16x16 equivalent

- `#define PMV_USESQUARES8 0x80000000`

see 16x16 equivalent

4.17.1 Define Documentation

4.17.1.1 `#define PMV_ADVANCEDDIAMOND8 0x00004000`

Uses advanced diamonds for 8x8 blocks.

Same as its 16x16 companion option

4.17.1.2 `#define PMV_EARLYSTOP16 0x00080000`

Dynamic ME thresholding.

PMVfast and EPZS stop search if current best is below some dynamic threshold. No diamond search is done, only halfpel refinement (if active). Without EARLYSTOP diamond search is always done. That would be much slower, but not really lead to better quality.

4.17.1.3 `#define PMV_EARLYSTOP8 0x08000000`

see 16x16 equivalent

Same as its 16x16 companion option

4.17.1.4 `#define PMV_EXTSEARCH16 0x00040000`

Extends search for 16x16 blocks.

Normal PMVfast predicts one start vector and does diamond search around this position. EXTSEARCH means that 2 more start vectors are used: (0,0) and median predictor and diamond search is done for those, too. Makes search slightly slower, but quality sometimes gets better.

4.17.1.5 `#define PMV_EXTSEARCH8 0x04000000`

see 16x16 equivalent

Same as its 16x16 companion option

4.17.1.6 `#define PMV_HALFPELDIAMOND16 0x00010000`

Turns on halfpel precision for 16x16 blocks.

switches the search algorithm from 1 or 2 full pixels precision to 1 or 2 half pixel precision.

4.17.1.7 #define PMV_HALFPELDIAMOND8 0x01000000

see 16x16 equivalent

Same as its 16x16 companion option

4.17.1.8 #define PMV_HALFPELREFINE16 0x00020000

Turns on halfpel refinement step.

After normal diamond search, an extra halfpel refinement step is performed. Should always be used if XVID_HALFPEL is on, because it gives a rather big increase in quality.

4.17.1.9 #define PMV_HALFPELREFINE8 0x02000000

see 16x16 equivalent

Same as its 16x16 companion option

4.17.1.10 #define PMV_OVERLAPPING16 0x00400000

Not implemented.

Same as above

4.17.1.11 #define PMV_OVERLAPPING8 0x40000000

see 16x16 equivalent

Same as its 16x16 companion option

4.17.1.12 #define PMV_QUICKSTOP16 0x00100000

Dynamic ME thresholding.

like EARLYSTOP, but not even halfpel refinement is done. Normally worse quality, so it defaults to off. Might be removed, too.

4.17.1.13 #define PMV_QUICKSTOP8 0x10000000

see 16x16 equivalent

Same as its 16x16 companion option

4.17.1.14 #define PMV_UNRESTRICTED16 0x00200000

Not implemented.

”unrestricted ME” is a feature of MPEG4. It’s not implemented, so this flag is ignored (not even checked).

4.17.1.15 #define PMV_UNRESTRICTED8 0x20000000

see 16x16 equivalent

Same as its 16x16 companion option

4.17.1.16 #define PMV_USESQUARES16 0x00800000

Use square pattern.

Replace the diamond search with a square search.

4.17.1.17 #define PMV_USESQUARES8 0x80000000

see 16x16 equivalent

Same as its 16x16 companion option

4.18 Encoder operations

4.18.1 Detailed Description

These are all the operations XviD's encoder can perform.

Defines

- `#define XVID_ENC_ENCODE 0`
Encodes a frame.
- `#define XVID_ENC_CREATE 1`
Creates a decoder instance.
- `#define XVID_ENC_DESTROY 2`
Destroys a encoder instance.

4.18.2 Define Documentation

4.18.2.1 `#define XVID_ENC_CREATE 1`

Creates a decoder instance.

This operation constant is used by a client application in order to create an encoder instance. Encoder instances are independant from each other.

4.18.2.2 `#define XVID_ENC_DESTROY 2`

Destroys a encoder instance.

This operation constant is used by the client application to destroy a previously created encoder instance.

4.18.2.3 `#define XVID_ENC_ENCODE 0`

Encodes a frame.

This operation constant is used when client application wants to encode a frame. Client application must also fill `XVID_ENC_FRAME` appropriately.

4.19 Encoder entry point

Functions

- `int xvid_encore (void *handle, int opt, void *param1, void *param2)`

Encoder entry point.

4.19.1 Function Documentation

4.19.1.1 `int xvid_encore (void * handle, int opt, void * param1, void * param2)`

Encoder entry point.

This is the XviD's encoder entry point. The possible operations are described in the [Encoder operations](#) section.

Parameters:

handle Encoder instance handle

opt Encoder option constant

param1 Used to pass [XVID_ENC_PARAM](#) or [XVID_ENC_FRAME](#) structures.

param2 Optionally used to pass the [XVID_ENC_STATS](#) structure.

Chapter 5

XviD core library Data Structure Documentation

5.1 XVID_ENC_FRAME Struct Reference

```
#include <xvid.h>
```

5.1.1 Detailed Description

Structure used to pass a frame to the encoder.

Data Fields

- int [general](#)
[in]
- int [motion](#)
[in]
- void * [bitstream](#)
[out]
- int [length](#)
[out]
- void * [image](#)
[in]
- int [colorspace](#)
[in]
- unsigned char * [quant_intra_matrix](#)
[in]

- unsigned char * [quant_inter_matrix](#)
[in]
- int [quant](#)
[in]
- int [intra](#)
[in/out]
- HINTINFO [hint](#)
[in/out]

5.1.2 Field Documentation

5.1.2.1 void* [XVID_ENC_FRAME::bitstream](#)

[out]

Output MPEG4 bitstream buffer pointer

5.1.2.2 int [XVID_ENC_FRAME::colorspace](#)

[in]

input frame colorspace

5.1.2.3 int [XVID_ENC_FRAME::general](#)

[in]

Sets general options flag (See [Flags for XVID_ENC_FRAME.general](#))

5.1.2.4 HINTINFO [XVID_ENC_FRAME::hint](#)

[in/out]

mv hint information

5.1.2.5 void* [XVID_ENC_FRAME::image](#)

[in]

Input frame

5.1.2.6 int [XVID_ENC_FRAME::intra](#)

[in/out]

- [in] : tells XviD if the frame must be encoded as an intra frame

- 1: forces the encoder to create a keyframe. Mainly used during a VBR 2nd pass.
 - 0: forces the encoder not to create a keyframe. Mainly used during a VBR second pass
 - -1: let the encoder decide (based on contents and max_key_interval). Mainly used in ABR mode and during a 1st VBR pass.
- [out] : When first set to -1, the encoder returns the effective keyframe state of the frame.

5.1.2.7 int XVID_ENC_FRAME::length

[out]

Output MPEG4 bitstream length (bytes)

5.1.2.8 int XVID_ENC_FRAME::motion

[in]

Sets Motion Estimation options

5.1.2.9 int XVID_ENC_FRAME::quant

[in]

Frame quantizer :

- 0 (zero) : Then the rate controller chooses the right quantizer for you. Typically used in ABR encoding, or first pass of a VBR encoding session.
- != 0 : Then you force the encoder to use this specific quantizer value. It is clamped in the interval [1..31]. Typically used during the 2nd pass of a VBR encoding session.

5.1.2.10 unsigned char* XVID_ENC_FRAME::quant_inter_matrix

[in]

Custom inter quantization matrix

5.1.2.11 unsigned char* XVID_ENC_FRAME::quant_intra_matrix

[in]

Custom intra quantization matrix

5.2 XVID_ENC_PARAM Struct Reference

```
#include <xvid.h>
```

5.2.1 Detailed Description

Structure used for encoder instance creation.

Data Fields

- int `width`
[in]
- int `height`
[in]
- int `fincr`
[in]
- int `fbase`
[in]
- int `rc.bitrate`
[in]
- int `rc.reaction_delay_factor`
[in]
- int `rc.averaging_period`
[in]
- int `rc.buffer`
[in]
- int `max_quantizer`
[in]
- int `min_quantizer`
[in]
- int `max_key_interval`
[in]
- void * `handle`
[out]

5.2.2 Field Documentation

5.2.2.1 int [XVID_ENC_PARAM::fbase](#)

[in]

Time base (fps = increment/base).

5.2.2.2 int [XVID_ENC_PARAM::fincr](#)

[in]

Time increment (fps = increment/base).

5.2.2.3 void* [XVID_ENC_PARAM::handle](#)

[out]

XviD core lib will set this with the creater encoder instance.

5.2.2.4 int [XVID_ENC_PARAM::height](#)

[in]

Input frame height.

5.2.2.5 int [XVID_ENC_PARAM::max_key_interval](#)

[in]

Sets the maximum interval between key frames.

5.2.2.6 int [XVID_ENC_PARAM::max_quantizer](#)

[in]

Sets the upper limit of the quantizer.

5.2.2.7 int [XVID_ENC_PARAM::min_quantizer](#)

[in]

Sets the lower limit of the quantizer.

5.2.2.8 int [XVID_ENC_PARAM::rc_averaging_period](#)

[in]

Tunes how fast the rate control reacts - lower values are faster.

5.2.2.9 int [XVID_ENC_PARAM::rc_bitrate](#)

[in]

Sets the target bitrate of the encoded stream, in bits/second. *

5.2.2.10 int [XVID_ENC_PARAM::rc_buffer](#)

[in]

Tunes how fast the rate control reacts - lower values are faster.

5.2.2.11 int [XVID_ENC_PARAM::rc_reaction_delay_factor](#)

[in]

Tunes how fast the rate control reacts - lower values are faster.

5.2.2.12 int [XVID_ENC_PARAM::width](#)

[in]

Input frame width.

5.3 XVID_ENC_STATS Struct Reference

```
#include <xvid.h>
```

5.3.1 Detailed Description

Encoding statistics.

Data Fields

- int [quant](#)
[out]
- int [hlength](#)
[out]
- int [kblks](#)
[out]
- int [mblks](#)
[out]
- int [ublks](#)
[out]

5.3.2 Field Documentation

5.3.2.1 int [XVID_ENC_STATS::hlength](#)

[out]

Header bytes in the resulting MPEG4 stream

5.3.2.2 int [XVID_ENC_STATS::kblks](#)

[out]

Number of intra macro blocks

5.3.2.3 int [XVID_ENC_STATS::mblks](#)

[out]

Number of inter macro blocks

5.3.2.4 int [XVID_ENC_STATS::quant](#)

[out]

Frame quantizer used during encoding

XviD API reference

5.3.2.5 int [XVID_ENC_STATS::ublks](#)

[out]

Number of skipped macro blocks

5.4 XVID_INIT_PARAM Struct Reference

```
#include <xvid.h>
```

5.4.1 Detailed Description

Structure used in xvid_init function.

Data Fields

- int [cpu_flags](#)
[in/out]
- int [api_version](#)
[out]
- int [core_build](#)
[out]

5.4.2 Field Documentation

5.4.2.1 int [XVID_INIT_PARAM::api_version](#)

[out]

xvid_init will initialize this field with the API_VERSION used in this XviD core library

5.4.2.2 int [XVID_INIT_PARAM::core_build](#)

[out]

Todo

Unused.

5.4.2.3 int [XVID_INIT_PARAM::cpu_flags](#)

[in/out]

Filled with desired[in] or available[out] cpu instruction sets.

Chapter 6

XviD core library Page Documentation

6.1 Todo List

Global [XVID_INIT_PARAM::core.build](#) Unused.

6.2 Deprecated List

Global **XVID_ALTERNATESCAN** This flag is no longer used.

Global **XVID_LATEINTRA** This flag is no longer used.

Global **XVID_LUMIMASKING** This flag is no longer used.

Global **XVID_TOPFIELDFIRST** This flag is no longer used.

Index

- API version, [8](#)
- api_grp
 - API_VERSION, [8](#)
- API_VERSION
 - api_grp, [8](#)
- api_version
 - XVID_INIT_PARAM, [41](#)
- bitstream
 - XVID_ENC_FRAME, [34](#)
- colorspace
 - XVID_ENC_FRAME, [34](#)
- Colorspaces constants., [10](#)
- core_build
 - XVID_INIT_PARAM, [41](#)
- cpu_flags
 - XVID_INIT_PARAM, [41](#)
- csp_grp
 - XVID_CSP_EXTERN, [10](#)
 - XVID_CSP_USER, [11](#)
- decentry_grp
 - xvid_decore, [21](#)
- Decoder entry point, [21](#)
- Decoder operations, [20](#)
- Decoder related functions and structures., [18](#)
- decops_grp
 - XVID_DEC_CREATE, [20](#)
 - XVID_DEC_DECODE, [20](#)
 - XVID_DEC_DESTROY, [20](#)
- encentry_grp
 - xvid_encore, [32](#)
- encgenflags_grp
 - XVID_ADAPTIVEQUANT, [24](#)
 - XVID_ALTERNATESCAN, [24](#)
 - XVID_CUSTOM_QMATRIX, [24](#)
 - XVID_GRAYSCALE, [24](#)
 - XVID_GREYSCALE, [24](#)
 - XVID_H263QUANT, [24](#)
 - XVID_HALFPPEL, [25](#)
 - XVID_HINTEDME_GET, [25](#)
 - XVID_HINTEDME_SET, [25](#)
 - XVID_INTER4V, [25](#)
 - XVID_INTERLACING, [25](#)
 - XVID_LATEINTRA, [25](#)
 - XVID_LUMIMASKING, [25](#)
 - XVID_ME_EPZS, [25](#)
 - XVID_ME_FULLSEARCH, [26](#)
 - XVID_ME_LOGARITHMIC, [26](#)
 - XVID_ME_PMVFAST, [26](#)
 - XVID_ME_ZERO, [26](#)
 - XVID_MPEGQUANT, [26](#)
 - XVID_TOPFIELDFIRST, [26](#)
- encmotionflags_grp
 - PMV_ADVANCEDDIAMOND8, [28](#)
 - PMV_EARLYSTOP16, [28](#)
 - PMV_EARLYSTOP8, [28](#)
 - PMV_EXTSEARCH16, [28](#)
 - PMV_EXTSEARCH8, [28](#)
 - PMV_HALFPPELDIAMOND16, [28](#)
 - PMV_HALFPPELDIAMOND8, [28](#)
 - PMV_HALFPPELREFINE16, [29](#)
 - PMV_HALFPPELREFINE8, [29](#)
 - PMV_OVERLAPPING16, [29](#)
 - PMV_OVERLAPPING8, [29](#)
 - PMV_QUICKSTOP16, [29](#)
 - PMV_QUICKSTOP8, [29](#)
 - PMV_UNRESTRICTED16, [29](#)
 - PMV_UNRESTRICTED8, [29](#)
 - PMV_USESQUARES16, [30](#)
 - PMV_USESQUARES8, [30](#)
- Encoder entry point, [32](#)
- Encoder operations, [31](#)
- Encoder related functions and structures., [22](#)
- encops_grp
 - XVID_ENC_CREATE, [31](#)
 - XVID_ENC_DESTROY, [31](#)
 - XVID_ENC_ENCODE, [31](#)
- Error codes returned by XviD API entry points., [9](#)
- error_grp
 - XVID_ERR_FAIL, [9](#)
 - XVID_ERR_FORMAT, [9](#)
 - XVID_ERR_MEMORY, [9](#)
 - XVID_ERR_OK, [9](#)
- fbase
 - XVID_ENC_PARAM, [37](#)
- fincr

- XVID_ENC_PARAM, 37
- Flags for XVID_DEC_FRAME.general, 19
- Flags for XVID_ENC_FRAME.general, 23
- Flags for XVID_ENC_FRAME.motion, 27
- Flags for XVID_INIT_PARAM.cpu_flags., 13
- general
 - XVID_ENC_FRAME, 34
- Global constants used in both encoder and decoder., 7
- handle
 - XVID_ENC_PARAM, 37
- height
 - XVID_ENC_PARAM, 37
- hint
 - XVID_ENC_FRAME, 34
- hlenth
 - XVID_ENC_STATS, 39
- ia64 specific cpu flags., 15
- ia64_grp
 - XVID_CPU_IA64, 15
- image
 - XVID_ENC_FRAME, 34
- inientry_grp
 - xvid_init, 17
- iniflags_grp
 - XVID_CPU_CHKONLY, 16
 - XVID_CPU_FORCE, 16
- Initialization commands., 16
- Initialization constants, structures and functions., 12
- Initialization entry point., 17
- intra
 - XVID_ENC_FRAME, 34
- kblks
 - XVID_ENC_STATS, 39
- length
 - XVID_ENC_FRAME, 35
- max_key_interval
 - XVID_ENC_PARAM, 37
- max_quantizer
 - XVID_ENC_PARAM, 37
- mblks
 - XVID_ENC_STATS, 39
- min_quantizer
 - XVID_ENC_PARAM, 37
- motion
 - XVID_ENC_FRAME, 35
- PMV_ADVANCEDDIAMOND8
 - encmotionflags_grp, 28
- PMV_EARLYSTOP16
 - encmotionflags_grp, 28
- PMV_EARLYSTOP8
 - encmotionflags_grp, 28
- PMV_EXTSEARCH16
 - encmotionflags_grp, 28
- PMV_EXTSEARCH8
 - encmotionflags_grp, 28
- PMV_HALFPELDIAMOND16
 - encmotionflags_grp, 28
- PMV_HALFPELDIAMOND8
 - encmotionflags_grp, 28
- PMV_HALFPELREFINE16
 - encmotionflags_grp, 29
- PMV_HALFPELREFINE8
 - encmotionflags_grp, 29
- PMV_OVERLAPPING16
 - encmotionflags_grp, 29
- PMV_OVERLAPPING8
 - encmotionflags_grp, 29
- PMV_QUICKSTOP16
 - encmotionflags_grp, 29
- PMV_QUICKSTOP8
 - encmotionflags_grp, 29
- PMV_UNRESTRICTED16
 - encmotionflags_grp, 29
- PMV_UNRESTRICTED8
 - encmotionflags_grp, 29
- PMV_USESQUARES16
 - encmotionflags_grp, 30
- PMV_USESQUARES8
 - encmotionflags_grp, 30
- quant
 - XVID_ENC_FRAME, 35
 - XVID_ENC_STATS, 39
- quant_inter_matrix
 - XVID_ENC_FRAME, 35
- quant_intra_matrix
 - XVID_ENC_FRAME, 35
- rc_averaging_period
 - XVID_ENC_PARAM, 37
- rc_bitrate
 - XVID_ENC_PARAM, 37
- rc_buffer
 - XVID_ENC_PARAM, 38
- rc_reaction_delay_factor
 - XVID_ENC_PARAM, 38
- ublks
 - XVID_ENC_STATS, 39
- width

- XVID_ENC_PARAM, 38
- x86 specific cpu flags, 14
- XVID_ADAPTIVEQUANT
 - encgenflags_grp, 24
- XVID_ALTERNATESCAN
 - encgenflags_grp, 24
- XVID_CPU_CHKONLY
 - iniflags_grp, 16
- XVID_CPU_FORCE
 - iniflags_grp, 16
- XVID_CPU_IA64
 - ia64_grp, 15
- XVID_CSP_EXTERN
 - csp_grp, 10
- XVID_CSP_USER
 - csp_grp, 11
- XVID_CUSTOM_QMATRIX
 - encgenflags_grp, 24
- XVID_DEC_CREATE
 - decops_grp, 20
- XVID_DEC_DECODE
 - decops_grp, 20
- XVID_DEC_DESTROY
 - decops_grp, 20
- xvid_decore
 - decentry_grp, 21
- XVID_ENC_CREATE
 - encops_grp, 31
- XVID_ENC_DESTROY
 - encops_grp, 31
- XVID_ENC_ENCODE
 - encops_grp, 31
- XVID_ENC_FRAME, 33
 - bitstream, 34
 - colospace, 34
 - general, 34
 - hint, 34
 - image, 34
 - intra, 34
 - length, 35
 - motion, 35
 - quant, 35
 - quant_inter_matrix, 35
 - quant_intra_matrix, 35
- XVID_ENC_PARAM, 36
 - fbase, 37
 - fincr, 37
 - handle, 37
 - height, 37
 - max_key_interval, 37
 - max_quantizer, 37
 - min_quantizer, 37
 - rc_averaging_period, 37
 - rc_bitrate, 37
 - rc_buffer, 38
 - rc_reaction_delay_factor, 38
 - width, 38
- XVID_ENC_STATS, 39
 - hlength, 39
 - kblks, 39
 - mblks, 39
 - quant, 39
 - ublks, 39
- xvid_encore
 - encentry_grp, 32
- XVID_ERR_FAIL
 - error_grp, 9
- XVID_ERR_FORMAT
 - error_grp, 9
- XVID_ERR_MEMORY
 - error_grp, 9
- XVID_ERR_OK
 - error_grp, 9
- XVID_GRAYSCALE
 - encgenflags_grp, 24
- XVID_GREYSCALE
 - encgenflags_grp, 24
- XVID_H263QUANT
 - encgenflags_grp, 24
- XVID_HALFPEL
 - encgenflags_grp, 25
- XVID_HINTEDME_GET
 - encgenflags_grp, 25
- XVID_HINTEDME_SET
 - encgenflags_grp, 25
- xvid_init
 - inientry_grp, 17
- XVID_INIT_PARAM, 41
 - api_version, 41
 - core_build, 41
 - cpu_flags, 41
- XVID_INTER4V
 - encgenflags_grp, 25
- XVID_INTERLACING
 - encgenflags_grp, 25
- XVID_LATEINTRA
 - encgenflags_grp, 25
- XVID_LUMIMASKING
 - encgenflags_grp, 25
- XVID_ME_EPZS
 - encgenflags_grp, 25
- XVID_ME_FULLSEARCH
 - encgenflags_grp, 26
- XVID_ME_LOGARITHMIC
 - encgenflags_grp, 26
- XVID_ME_PMVFAST
 - encgenflags_grp, 26

XVID_ME_ZERO
 encgenflags_grp, [26](#)
XVID_MPEGQUANT
 encgenflags_grp, [26](#)
XVID_TOPFIELDFIRST
 encgenflags_grp, [26](#)