

## XviD API 2.1 Reference (for 0.9.x series)

Author: XviD Team

2002-11-24



---

# Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>XviD core library Module Index</b>                       | <b>1</b> |
| 1.1      | XviD core library Modules . . . . .                         | 1        |
| <b>2</b> | <b>XviD core library Data Structure Index</b>               | <b>3</b> |
| 2.1      | XviD core library Data Structures . . . . .                 | 3        |
| <b>3</b> | <b>XviD core library Page Index</b>                         | <b>5</b> |
| 3.1      | XviD core library Related Pages . . . . .                   | 5        |
| <b>4</b> | <b>XviD core library Module Documentation</b>               | <b>7</b> |
| 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       |

---

|          |   |           |
|----------|---|-----------|
| <b>5</b> | <b>XviD core library Data Structure Documentation</b> | <b>33</b> |
| 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        |
| <b>6</b> | <b>XviD core library Page Documentation</b>           | <b>43</b> |
| 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:

|  |    |
|--|----|
| <a href="#">XVID_ENC_FRAME</a> (Structure used to pass a frame to the encoder) . . . . . | 33 |
| <a href="#">XVID_ENC_PARAM</a> (Structure used for encoder instance creation) . . . . .  | 36 |
| <a href="#">XVID_ENC_STATS</a> (Encoding statistics) . . . . .                           | 39 |
| <a href="#">XVID_INIT_PARAM</a> (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 . . . . .       | <a href="#">43</a> |
| Deprecated List . . . . . | <a href="#">44</a> |



---

## 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 API\_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 API_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 successfully. 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 exeperi-  
mental 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\\_HALFPPEL](#) 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 adaptative 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 flags 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

**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\_I420, [10](#)
    - XVID\_CSP\_NULL, [10](#)
    - XVID\_CSP\_RGB24, [10](#)
    - XVID\_CSP\_RGB32, [10](#)
    - XVID\_CSP\_RGB555, [10](#)
    - XVID\_CSP\_RGB565, [10](#)
    - XVID\_CSP\_USER, [11](#)
    - XVID\_CSP\_UYVY, [10](#)
    - XVID\_CSP\_VFLIP, [10](#)
    - XVID\_CSP\_YUY2, [10](#)
    - XVID\_CSP\_YV12, [10](#)
    - XVID\_CSP\_YVYU, [10](#)
  - decentry\_grp
    - xvid\_deccore, [21](#)
  - decframe\_grp
    - XVID\_QUICK\_DECODE, [19](#)
  - 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](#)
    - XVID\_VALID\_FLAGS, [23](#)
  - encmotionflags\_grp
    - PMV\_ADVANCEDDIAMOND16, [27](#)
    - 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
- hlength
  - 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\_ADVANCEDDIAMOND16
  - encmotionflags\_grp, 27
- 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
- x86\_grp
  - XVID\_CPU\_3DNOW, 14
  - XVID\_CPU\_3DNOWEXT, 14
  - XVID\_CPU\_MMX, 14
  - XVID\_CPU\_MMXEXT, 14
  - XVID\_CPU\_SSE, 14
  - XVID\_CPU\_SSE2, 14
  - XVID\_CPU\_TSC, 14
- XVID\_ADAPTIVEQUANT
  - encgenflags\_grp, 24
- XVID\_ALTERNATESCAN
  - encgenflags\_grp, 24
- XVID\_CPU\_3DNOW
  - x86\_grp, 14
- XVID\_CPU\_3DNOWEXT
  - x86\_grp, 14
- XVID\_CPU\_CHKONLY
  - iniflags\_grp, 16
- XVID\_CPU\_FORCE
  - iniflags\_grp, 16
- XVID\_CPU\_IA64
  - ia64\_grp, 15
- XVID\_CPU\_MMX
  - x86\_grp, 14
- XVID\_CPU\_MMXEXT
  - x86\_grp, 14
- XVID\_CPU\_SSE
  - x86\_grp, 14
- XVID\_CPU\_SSE2
  - x86\_grp, 14
- XVID\_CPU\_TSC
  - x86\_grp, 14
- XVID\_CSP\_EXTERN
  - csp\_grp, 10
- XVID\_CSP\_I420
  - csp\_grp, 10
- XVID\_CSP\_NULL
  - csp\_grp, 10
- XVID\_CSP\_RGB24
  - csp\_grp, 10
- XVID\_CSP\_RGB32
  - csp\_grp, 10
- XVID\_CSP\_RGB555
  - csp\_grp, 10
- XVID\_CSP\_RGB565
  - csp\_grp, 10
- XVID\_CSP\_USER
  - csp\_grp, 11
- XVID\_CSP\_UYVY
  - csp\_grp, 10
- XVID\_CSP\_VFLIP
  - csp\_grp, 10
- XVID\_CSP\_YUY2
  - csp\_grp, 10
- XVID\_CSP\_YV12
  - csp\_grp, 10
- XVID\_CSP\_YVYU
  - csp\_grp, 10
- 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
  - decetry\_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
  - mbks, 39
  - quant, 39
  - ubks, 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\_HALFPPEL
  - 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\_QUICK\_DECODE
  - decframe\_grp, 19
- XVID\_TOPFIELDFIRST
  - encgenflags\_grp, 26
- XVID\_VALID\_FLAGS
  - encgenflags\_grp, 23