



# **DivX Encoder API Settings**

## **Functional Specification**

**Release Date: August 25, 2005**

This document contains proprietary information that is protected by copyright. No part of this document may be photocopied, reproduced, or translated without the prior written consent of DivX, Inc. The information contained in this document is subject to change without notice.

**DIVX, INC. CONFIDENTIAL  
DO NOT COPY**

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>ii</b>
1.1	Understanding Parameters for this API.....	1
<b>2</b>	<b>Parameters in Detail.....</b>	<b>2</b>
2.1	General Parameters.....	2
2.2	Preprocessor Parameters .....	3
2.3	Rate Control and Frame Decision Parameters.....	6
2.4	Core Encoder Parameters.....	8
<b>Appendix A</b>	<b>Command Line Interpreter (CLI).....</b>	<b>11</b>
2.5	CLI Syntax.....	11
2.6	Resize Combo Argument.....	13

# Document History

Document Version	Product Version	File Name	Release Date
1 Prerelease	DivX 6 CE SDK RC-1	fs_divx6_encoderapisettings.pdf	04/15/05
1	DivX 6 CE SDK V3.0	fs_divx6_encoderapisettings.pdf	06/17/05
1.1	DivX 6 CE SDK V3.0.1	fs_divx6_encoderapisettings.pdf	08/25/05
1.2	DivX 6 For Linux	fs_divx6_encoderapisettings.pdf	01/26/06

# Document Change Log

Section Number	Page Number(s)	Description of Change
N/A	ii-1	Updated publication date and Document History

# 1 Introduction

This document describes the various encoder parameters for the Encoder API delivered with this SDK. This document is designed to be used in tandem with the *Using the Encoder API Application Note* included in this documentation set.

## 1.1 Understanding Parameters for this API

The parameters for this API are set by the user at initialization time or, in some cases, during encoding. The Settings API supports the six types of parameters listed below:

- **Boolean:** setting has two possible values, true or false.
- **Enumerated:** parameter can take one of a predefined set of values
- **Integer:** numeric parameter accepting integer values
- **Double:** numeric parameter accepting a range of real values
- **String:** 8-bit ASCII string parameter
- **Data:** parameter is a binary data block

The parameters listed above are detailed in subsequent sections of this functional specification.

The user should query the Settings API to determine if a parameter is enabled or read-only. Parameters may also have constraints imposed by their implementation, e.g. minimum and maximum values. Such constraints are not within the scope of this document.

Parameters may have dependencies on one another. Changing one parameter may have an effect on another parameter. Dependencies are managed in the Settings API by ordering the parameters in order of dependency. This guarantees that manipulating one parameter will not affect any higher-precedence parameter. It could, however, affect the value, range or status of a parameter of lower precedence. For guidance in settings for this API, refer to the *Using the Encoder API Application Note* included in this documentation set.

## 2 Parameters in Detail

This section organizes parameters and their descriptions as follows:

- General Parameters
- Preprocessor Parameters
- Rate Control and Frame Decision Parameters
- Core Encoder Parameters

### 2.1 General Parameters

This section discusses general parameters.

#### 2.1.1 Enumerated: "profile"

This parameter controls the encoder's certification profile. One effect of this parameter is the enforcement of certain constraints upon other parameters (e.g. rate control parameters). Values are described in the table below.

"profile" value	Integer Value	Notes
PROFILE_HANDHELD	2	The encoder will generate a bitstream compatible with the Handheld DivX Certification Profile
PROFILE_PORTABLE	4	The encoder will generate bitstream compatible with the Portable DivX Certification Profile
PROFILE_HOME_THEATER	8	The encoder will generate bitstream compatible with the Home Theater/ Ultra Home Theater DivX Certification Profile
PROFILE_HDTV	16	The encoder will generate bitstream compatible with the High Definition DivX Certification Profile
PROFILE_UNCONSTRAINED	1	The encoder will not enforce or respect any DivX Certification Profile.

#### 2.1.2 Enumerated: "encoder"

This parameter may be used to set the encoder's output format. Values are described in the table below.

"encoder" value	Integer Value	Notes
ENCODER_MPEG4	16	The encoder's output format will be DivX 5 bitstream (MPEG-4 compatible).

#### 2.1.3 String: "working\_folder"

This should be set to a string containing an absolute file system path.

## 2.2 Preprocessor Parameters

This section describes parameters that control the encoder's preprocessing tasks.

### 2.2.1 Boolean: "timescale\_override"

This parameter affects the timescale (or 'time\_increment\_resolution' in MPEG-4 parlance) of the output video. Its behavior is described in the following table:

"timescale_override" value	Action
false	The encoder's output timescale will be identical to the input video's timescale as specified in <code>FormatInfo::timescale</code> when <code>IQEncoderInterface::setFormatInput()</code> was called.
true	The encoder's output timescale will be set equal to the value of integer parameter "timescale" (see 2.2.2). The timestamp of each encoded video frame will be rescaled accordingly.

### 2.2.2 Integer: "timescale"

The timescale of the output video (or 'time\_increment\_resolution' in MPEG-4 parlance) may be controlled using this parameter (see above).

### 2.2.3 Integer: "crop\_left"

This specifies the number of pixels that will be cropped from the left edge of the input video.

### 2.2.4 Integer: "crop\_right"

This specifies the number of pixels that will be cropped from the right edge of the input video.

### 2.2.5 Integer: "crop\_top"

This specifies the number of pixels that will be cropped from the top edge of the input video.

### 2.2.6 Integer: "crop\_bottom"

This specifies the number of pixels that will be cropped from the bottom edge of the input video.

**2.2.7 Enumerated: "resize\_mode"**

This parameter is the master control for the preprocessor's resize engine. This parameter's allowed values are listed in the following table.

"resize_mode" value	Integer Value	Notes
RESIZE_MODE_OFF	1	No resizing will be performed.
RESIZE_MODE_ON	4	The image will be resized to the dimensions specified by parameters "resize_width" and "resize_height".
RESIZE_MODE_AUTO_HEIGHT	2	The image will be resized so that it respects the pixel aspect ratios of the input and output video. The width of the output video will be set by the value of "resize_width". The height of the output video will be calculated to minimize aspect distortions in the output video. Output height is a function of output width, input dimensions, input and output PAR. Height is rounded to the closest non-zero multiple of 16.

**2.2.8 Enumerated: "resize\_algorithm"**

This parameter specifies the filter kernel that will be used to resize the video. Allowable values are listed below.

"resize_algorithm" value	Integer Value	Notes
RESIZE_ALGORITHM_BILINEAR	1	A 2-tap bilinear filter.
RESIZE_ALGORITHM_BICUBIC_SOFT	2	A simple bicubic algorithm with parameters: b=0.333, c=0.333
RESIZE_ALGORITHM_BICUBIC_NORMAL	4	A simple bicubic algorithm with parameters: b=0.0, c=0.5
RESIZE_ALGORITHM_BICUBIC_SHARP	8	A simple bicubic algorithm with parameters: b=0.0, c=0.75

**2.2.9 Integer: "resize\_width"**

Parameter "resize\_width" specifies the desired video output horizontal dimension. Note that in addition to minimum and maximum values, there may be additional restrictions on the value of this parameter. Specifically, if a resize width that is not a multiple of 4 is set, encoding may fail.

**2.2.10 Integer: "resize\_height"**

Parameter "resize\_height" specifies the desired video output vertical dimension. Note that in addition to minimum and maximum values, there may be additional restrictions on the value of this parameter. Specifically, if a resize width that is not a multiple of 2 is set, encoding may fail.

**2.2.11 Enumerated: "par\_input"**

This parameter specifies the pixel-aspect-ratio of the video that is input to the encoder. . Possible values are listed below.

"par_input" value	Notes
PAR_1_1	Input pixels are square
PAR_12_11	Input PAR is that of 625-line 4:3
PAR_10_11	Input PAR is that of 525-line 4:3
PAR_16_11	Input PAR is that of 625-line 16:9
PAR_40_33	Input PAR is that of 525-line 16:9
PAR_AUTO	Input pixel aspect ratio should be taken from FormatInfo::pixelAspectX and FormatInfo::pixelAspectY as passed to setFormatInput()
PAR_UNKNOWN	Input pixel aspect ratio is not known

**2.2.12 Enumerated: "par\_output"**

This parameter specifies the pixel-aspect-ratio of the video that will be output from the encoder. This value will be encoded into the VOL header of the output bitstream. Possible values are listed below.

"par_input" value	Notes
PAR_1_1	Output pixels are square
PAR_12_11	Output PAR is that of 625-line 4:3
PAR_10_11	Output PAR is that of 525-line 4:3
PAR_16_11	Output PAR is that of 625-line 16:9
PAR_40_33	Output PAR is that of 525-line 16:9
PAR_AUTO	Output PAR should be computed as a function of input PAR after taking into account the effect of any resize operation performed by the pre-processor.
PAR_AUTO_NOT_EXT	As PAR_AUTO but the encoder will round the PAR to the nearest preset value. (one of 1:1, 12:11, 10:11, 16:11, 40:33). In this mode, extended PAR is never inserted into the bitstream.

**2.2.13 Enumerated: "filter\_mode"**

This parameter specifies the mode of the preprocessor's noise reduction filter. Possible values are listed below.

"filter_mode" value	Integer Value	Notes
FILTER_MODE_OFF	2	No noise reduction will be applied.
FILTER_MODE_LIGHT	4	The noise reduction filter will be enabled, with a preset strength of 'light'.
FILTER_MODE_NORMAL	8	The noise reduction filter will be enabled, with a preset strength of 'normal'.
FILTER_MODE_STRONG	16	The noise reduction filter will be enabled, with a preset strength of 'strong'.
FILTER_MODE_EXTREME	32	The noise reduction filter will be enabled, with a preset strength of 'extreme'.
FILTER_MODE_CUSTOM	64	The noise reduction filter will be enabled with custom settings. See parameters "filter_temporal", "filter_temporal_strength", "filter_spatial_passes", "filter_spatial_strength" for more information.



**2.2.14 Boolean: "filter\_temporal"**

Enables or disables the temporal element of the preprocessor's noise reduction filter. Set 'true' to enable the temporal filter.

**2.2.15 Double: "filter\_temporal\_strength"**

This parameter controls the strength of the temporal element of the preprocessor's noise reduction filter.

"filter_temporal_strength" value	Notes
0.0	Filter has no effect.
1.0	Maximum filter strength.

**2.2.16 Integer: "filter\_spatial"**

Parameter "filter\_spatial" controls the number of passes performed by the preprocessor's spatial noise reduction filter. A value of "0" will disable the spatial filter.

**2.2.17 Double: "filter\_spatial\_strength"**

This parameter controls the strength of the spatial element of the preprocessor's noise reduction filter.

"filter_spatial_strength" value	Notes
0.0	Filter has no effect.
1.0	Maximum filter strength.

**2.2.18 Boolean: "deinterlace"**

When set to True", the preprocessing deinterlaces the input video as a first step.

**2.2.19 Boolean: "deinterlace\_top\_field"**

This parameter selects a "master" field for the de-interlace algorithm. When set to "True", the top field is master. When set to "False", the bottom field is master. The temporal position of each output frame will be that of its input master field. For most applications, this distinction is unimportant. It has the effect of time-shifting interlaced input by one field period.

**2.3 Rate Control and Frame Decision Parameters**

This section describes parameters that control the encoder's rate control and frame decision.

**2.3.1 Enumerated: "rcmode"**

This parameter specifies the rate control mode. Possible values are listed below.

Integer Value	"rcmode" value
1	RCMODE_1PASS_CONSTANT_Q
2	RCMODE_VBV_1PASS
4	RCMODE_VBV_MULTIPASS_1ST
8	RCMODE_VBV_MULTIPASS_NTH

**2.3.2 Integer: "bitrate"**

Integer parameter "bitrate" controls the target bitrate of the encoder's rate control. Its units are bits-per-second.

**2.3.3 Double: "quantizer"**

This parameter specifies the desired encoding quantization parameter, or "QP". When this parameter takes a value that is not an integer, the encoder's rate control dithers individual integer frame quantization parameters to achieve the desired result.

**2.3.4 Integer: "vbv\_bitrate"**

This specifies the channel bitrate to be used in the rate control's VBV buffer model. Its units are bits-per-second. For consistency with the MPEG-4 specification, it is recommended to use a multiple of 400.

**2.3.5 Integer: "vbv\_size"**

Integer parameter "vbv\_size" sets the size in bits of the buffer modeled by the rate control's VBV buffer model. For consistency with the MPEG-4 specification, it is recommended to use a multiple of 16384.

**2.3.6 Integer: "vbv\_occupancy"**

Integer parameter "vbv\_occupancy" sets the initial fullness of the VBV buffer. Its units are bits. For consistency with the MPEG-4 specification, it is recommended to use a multiple of 64.

**2.3.7 Integer: "max\_b\_frames"**

The integer parameter "max\_b\_frames" allows the user to control the maximum permissible number of consecutive bi-directionally predicted video frames ("B-frames"). Setting this parameter to zero guarantees that no B-frames will be generated by the encoder. Because the B-frame insertion algorithm is adaptive, this parameter is a maximum constraint. There is no guarantee that the specified number of consecutive B-frames will occur in any particular given video sequence.

**2.3.8 Integer: "max\_key\_interval"**

Integer parameter "max\_key\_interval" may be used to control the maximum number of inter frames that can be encoded before an intra frame is mandated.

**2.3.9 Integer: "key\_frame\_threshold"**

This parameter allows the user to bias the encoder's I/P frame type decision. Legal values are in the range 0 to 100 inclusive. The encoder's scene change decision algorithms have been designed such that a key\_frame\_threshold of 50 will produce optimal encoding results.

<b>"key_frame_threshold" value</b>	<b>Notes</b>
0	Every frame will be encoded as an I-frame.
50	Optimal setting for I/P decision.
100	No frames will be encoded as I until either a frame in the key-list is encountered or the maximum key frame interval is reached.

**2.3.10 String: "keylist"**

This parameter allows the user to provide a list of frame sequence number that must be encoded as intra- or key-frames. The following two alternate syntaxes are available:

- Delimited list of integers, e.g. "100,200,300" or "100 200 300"
- Filename containing a delimited list of integers, e.g. "keylist.txt"

**2.3.11 String: "log\_file"**

Controls the filename of the logfile used to store video characteristics between passes in a multipass encode.

**2.3.12 Boolean: "log\_file\_write"**

The behavior of this parameter is described in the following table:

"log_file_write"	Notes
false	No log file will be written or modified. Note that when "rcmode" is set to RCMODE_VBV_MULTIPASS_1ST, a log file will always be created.
true	The log file specified by "log_file" will be written. If the file already exists, and "rcmode" is set to RCMODE_VBV_MULTIPASS_NTH the data already in the file will be used as a base from which the new file's frame parameters are generated.

**2.3.13 Double: "complexity\_modulation"**

When available, "complexity\_modulation" allows the user to bias the bit distribution between complex and simple frames. A complex scene is one containing a great deal of motion and/or complex texture.

"complexity_modulation"	Notes
negative	Assigns more bits to low-complexity scenes and less to high-complexity scenes. A larger magnitude value will have a greater effect.
0	Neutral bit distribution between simple and complex scenes.
positive	Assigns more bits to high-complexity scenes and less to low-complexity scenes. A larger magnitude value will have a greater effect.

**2.4 Core Encoder Parameters**

This section describes parameters that control the core encoder.

**2.4.1 Integer: "performance"**

This allows the user to choose a trade-off between encoder CPU usage and encoder quality performance. A higher value of "performance" will produce better results, but will result in longer encoding times.

The "performance" variable is approximately proportional to log (encoding time). In the VFW and other GUIs, recommended preset values of performance are used: these are sweet-spots that have been tested and tuned. In between these presets, encoding time and quality should

increase monotonically, though not always constantly as certain features and advanced algorithms are switched on at different thresholds.

This parameter can bias against (or even disable) slower algorithms such as B-frames if necessary to achieve improved speed. As far as possible, and regardless of other encoder features, this has been implemented such that encoding time is proportional to  $k^{\text{performance}}$ .

Future releases will provide the ability to adjust this parameter while encoding is in progress. This could be done as part of a feedback loop that monitors current CPU usage: to achieve optimal usage of available CPU for real-time encoding.

"performance"	Notes
6950	Encoding time approximately equivalent to DivX 5.2 Fastest mode
7270	Encoding time approximately equivalent to DivX 5.2 Fast mode (all advanced features disabled)
7670	Encoding time approximately equivalent to DivX 5.2 Standard mode (all advanced features disabled)
8100	Encoding time approximately equivalent to DivX 5.2 Slow mode (all advanced features disabled)
8610	Encoding time approximately equivalent to DivX 5.2 Very Slow mode (all advanced features disabled)

#### 2.4.2 Boolean: "interlace"

This parameter controls whether the encoder generates an interlaced bitstream. When set to "True," the interlaced bitstream syntax is enabled.

#### 2.4.3 Boolean: "top\_field\_first"

This parameter is used to specify the field order of the input video frames. It controls the value of its namesake MPEG-4 header field "top\_field\_first".

#### 2.4.4 Boolean: "use\_gmc"

This parameter controls whether the encoder generates a bitstream that requires global-motion compensation, or GMC, at the decoder. When set true, GMC is enabled.

#### 2.4.5 Boolean: "quarter\_pel"

This parameter controls whether the encoder uses quarter-pixel motion estimation and compensation. When set to "True", quarter-pixel is enabled, requiring a decoder that supports it. When set to "False", half-pixel motion estimation / compensation is used instead.

#### 2.4.6 Boolean: "allow\_inter4v"

This parameter controls whether or not the encoder is permitted to make use of the INTER4V motion compensation mode. When set to "True" the encoder may use INTER4V, although the actual use of INTER4V macroblocks depends on the video source and other encoder settings. When set to "False", INTER4V will not be used.

#### 2.4.7 Enumerated: "quantization"

This parameter specifies the quantization algorithm to be used by the encoder.

"quantization" value	Integer Value	Notes
QUANTIZATION_H263	2	A basic H.263 quantization algorithm will be used.
QUANTIZATION_H263_OPT	4	A H.263 quantization algorithm optimized for PSNR will be used.
QUANTIZATION_MPEG	8	A basic MPEG quantization algorithm will be used.

#### 2.4.8 Enumerated: "psychovisual"

This parameter specifies the psychovisual enhancement algorithm to be used by the encoder.

"psychovisual" value	Integer Value	Notes
PSYCHOVISUAL_OFF	2	No psychovisual enhancements will be performed by the encoder. For best PSNR, select this mode.
PSYCHOVISUAL_SHAPING	4	Selects the noise-shaping psychovisual algorithm (fast)
PSYCHOVISUAL_MASKING	8	Selects the noise-masking psychovisual algorithm (slow)

#### 2.4.9 Integer: "data\_partitioning"

When "data\_partitioning" is set to "0", data partitioning is disabled and will not be present in the output bitstream. A non-zero value will enable data partitioning and control the maximum number of data bits per packet.

## Appendix A Command Line Interpreter (CLI)

This section describes the functionality of the Command Line Interpreter (CLI) used in the DivX 6 Encoder.

The CLI is a mechanism that allows any configuration of the encoder settings to be represented as an ASCII string. A simple API is provided to enable the user to perform the following tasks:

- produce a CLI string that represents the current encoder settings
- set multiple encoder parameters using information contained in a CLI string

The mapping of parameters to CLI constructs is described in the following subsections.

### 2.5 CLI Syntax

A CLI string is an 8-bit ASCII null-terminated string. It is made up of one or more arguments. Each argument begins with the character '-'. If a CLI string contains multiple arguments, these are separated by a single space.

#### 2.5.1 Simple Boolean Parameters

The Boolean parameters listed in the following table have a simple representation on the CLI. The presence or absence of the CLI argument controls the value of the parameter.

Parameter name	CLI argument for true value	CLI argument for false value
"quarter_pel"	-q	N/A
"use_gmc"	-g	N/A
"log_file_write "	-w	N/A
"use_dialogs "	-p	N/A
"enable_feedback "	N/A	-nf

#### 2.5.2 Simple Integer Parameters

The Integer parameters listed in the following table have a simple representation on the CLI. "x" represents the value of the parameter. The parameter will be assigned a default value if the argument is missing from the string.

Parameter name	CLI argument
"performance"	-pq x
"max_b_frames"	-b x
"key_frame_threshold"	-sc x
"max_key_interval"	-key x
"data_partitioning"	-dp x

#### 2.5.3 Simple String Parameters

The Integer parameters listed in the following table have a simple representation on the CLI. The value of the string is represented by <string>. If the string contains spaces, it must be enclosed in quotes "like this". If its argument is missing from the CLI string, the parameter will be assigned a default value.

Parameter name	CLI argument
"keylist"	-keylist <string>
"log_file"	-log_file <string>
"working_folder"	-dir <string>

### 2.5.4 Pixel Aspect Ratio Parameters

The following enumerated parameters have a special representation on the CLI string designed for conveying a pixel aspect ratio:

Parameter name	CLI argument
"par_input"	-parin <PAR>
"par_output"	-parout <PAR>

The pixel aspect ratio is represented within <PAR> as shown below:

<PAR>	Parameter value
1:1	PAR_1_1
12:11	PAR_12_11
10:11	PAR_10_11
16:11	PAR_16_11
40:33	PAR_40_33
auto	PAR_AUTO
autopreset	PAR_AUTO_NOT_EXT
unknown	PAR_UNKNOWN

### 2.5.5 Rate Control Combo Argument

Multiple API parameters are represented within a single CLI argument as shown below:

CLI argument	Value of enumerated parameter "rcmode"	Name of integer value 'x'
-bv1 x	RCMODE_VBV_1PASS	"bitrate"
-b1q x	RCMODE_1PASS_CONSTANT_Q	"quantizer"
-bvn1 x	RCMODE_VBV_MULTIPASS_1ST	"bitrate"
-bvnn x	RCMODE_VBV_MULTIPASS_NTH	"bitrate"

For example, when parsing "-b1q 12", enumerated parameter "rcmode" would be assigned value RCMODE\_1PASS\_CONSTANT\_Q and the integer parameter "quantizer" would be set to "12".

### 2.5.6 VBV Combo Argument

Three integer API parameters are represented within a single CLI argument as shown below:

`-vbv <vbv_bitrate>,<vbv_size>,<vbv_occupancy>`

If this argument is not present on the CLI, default values will be used instead.

### 2.5.7 Crop Combo Argument

Four integer API parameters are represented within a single CLI argument thus:

`-c <crop_left>,< crop_right>,< crop_top >,< crop_bottom >`

If this argument is not present on the CLI, then default values will be used instead.

## 2.6 Resize Combo Argument

A single CLI argument contains four API parameters. The presence and the format of this argument are controlled by enumerated parameter "resize\_mode", as shown in the following table.

Value of "resize_mode"	Argument format
RESIZE_MODE_OFF	<i>argument not present</i>
RESIZE_MODE_ON	-r <resize_width>, <resize_height>, <algorithm>
RESIZE_MODE_AUTOHEIGHT	-r <resize_width>, 0, <algorithm>

The fields <resize\_width> and <resize\_height> represent the values of integer parameters "resize\_width" and "resize\_height" respectively.

The field <algorithm> is a representation of enumerated parameter "resize\_algorithm", as shown in the following table:

Value of "resize_algorithm"	<algorithm>
RESIZE_ALGORITHM_BILINEAR	1
RESIZE_ALGORITHM_BICUBIC_SOFT	2
RESIZE_ALGORITHM_BICUBIC_NORMAL	3
RESIZE_ALGORITHM_BICUBIC_SHARP	4

### 2.6.1 Preprocess Combo Argument

Up to five API parameters are represented within a single CLI argument. The presence and the format of this argument depend on the value of enumerated parameter "filter\_mode", as shown in the table below.

Value of "filter_mode"	Argument format
FILTER_MODE_OFF	<i>argument not present</i>
FILTER_MODE_LIGHT	-pre 1
FILTER_MODE_NORMAL	-pre 2
FILTER_MODE_STRONG	-pre 3
FILTER_MODE_EXTREME	-pre 4
FILTER_MODE_CUSTOM	-pre <filter_temporal>, <filter_temporal_strength>, <filter_spatial>, <filter_spatial_strength>

✖ **Note:** If present, boolean parameter "filter\_temporal" is shown as '1' for true and '0' for false.

### 2.6.2 Arguments for Other Parameters

For parameters not represented in CLI arguments described above, there is a default representation for each type of parameter.

Parameter type	Parameter value	CLI argument
Boolean	false	-<parameter name>=0
	true	-<parameter name>=1
Enumerated	1<<x	-<parameter name>=x
Integer	x	-<parameter name>=x
Double	x	-<parameter name>=x
String	<string>	-<parameter name>=<string>
Data	<data>	-<parameter name>=<data>

Data parameters are conveyed on the CLI in their Base64 representation.