FFmpeg X264编码参数

目录

[1. 码率控制 1](#_Toc354826885)

[2. X264的preset和tune 2](#_Toc354826886)

[3. 编码延时建议 2](#_Toc354826887)

[4. ffmpeg编码参数和x264参数对照 3](#_Toc354826888)

[5. x264参数说明 5](#_Toc354826889)

# 码率控制

X264提供三种码率控制的方式：bitrate, qp, crf。这三种方式是互斥的，使用时设置其中之一即可。

（1）bitrate

x264会尝试把给定的位元率作为整体平均值来编码。这意味着最终编码文件的大小是已知的，但最终的品质未知。此选项通常与-pass（两阶段编码）一起使用。

注意，ffmpeg中设置bitrate的具体参数为bit\_rate，单位是bits/s（x264里面对应参数i\_bitrate的单位则是kbits/s, 1kbits为1000bits而非1024bits）.

（2）qp

使用qp选项时，表示P帧的量化值为qp。I帧和B帧的量化值则是从--ipratio和--pbratio中取得。可使用默认参数，也可自己设定。使用qp模式，即固定量化值，意味着停用弹性量化（aq\_mode）。

当qp为0时，为无损编码。

（3）crf

固定位元率系数，**C**onstant **R**ate**f**actor，

可用的值从1到51，越小编码质量越好，码率越高。一般使用16到24，可以为浮点。(crf并不是恒定质量的方式，同一片子同一crf值，其他参数不同可能码率和质量差较大，不同的片子之间就更没有可比性了)

此模式把某个“质量”作为编码目标，根据片子质量自动分配码率的vbr（Variable Bit Rate动态比特率）。X264中构想是让crf n提供与qp n相当的视觉品质，但编码文件更小一些。

CRF是借由降低“较不重要”帧的品质来达到此目的。在此情况下，“较不重要”是指在复杂或高动态场景的帧，其品质不是很耗费位元数就是不易察觉，所以会提高它们的量化值。从这些帧里所节省下来的位元数被重新分配到可以更有效利用的帧。

当crf为0时，与qp 0相同，实现无损编码。

# X264的preset和tune

鉴于x264的参数众多，各种参数的配合复杂，为了使用者方便，x264建议如无特别需要可使用preset和tune设置。这套开发者推荐的参数较为合理，可在此基础上在调整一些具体参数以符合自己需要，手动设定的参数会覆盖preset和tune里的参数。

--preset的参数主要调节编码速度和质量的平衡，有ultrafast、superfast、veryfast、faster、fast、medium、slow、slower、veryslow、placebo这10个选项，从快到慢。

--tune的参数主要配合视频类型和视觉优化的参数，或特别的情况。如果视频的内容符合其中一个可用的调整值又或者有其中需要，则可以使用此选项，否则建议不使用（如tune grain是为高比特率的编码而设计的）。tune的值有：

film： 电影、真人类型；

animation： 动画；

grain： 需要保留大量的grain时用；

stillimage： 静态图像编码时使用；

psnr： 为提高psnr做了优化的参数；

ssim： 为提高ssim做了优化的参数；

fastdecode： 可以快速解码的参数；

zerolatency：零延迟，用在需要非常低的延迟的情况下，比如电视电话会议的编码。

# 一些编码建议

## 编码延时

降低x264的延时是可能的，但是会降低质量。若需零延时，设置--tune zerolatency。若你可以接受一点儿小延时(如小于1秒)，最好还是允许延时。下列步骤可以降低延迟，当延迟足够小时，就别再进行后续步骤了：

1. 从初始值开始
2. 关闭sync-lookahead(设置用于线程预测的帧缓存大小。最大值是250.在第二遍及更多遍编码或基于分片线程时自动关闭)
3. 降低rc-lookahead，但别小于10(设定mb-tree位元率控制和vbv-lookahead使用的帧数)
4. 降低threads(比如从12降到6)
5. 使用切片线程(sliced threads)
6. 禁用rc-lookahead
7. 禁用b-frames
8. 实在不行，就用--tune zerolatency

param->rc.i\_lookahead = 0;param->i\_sync\_lookahead = 0; param->i\_bframe = 0; param->b\_sliced\_threads = 1; param->b\_vfr\_input = 0;param->rc.b\_mb\_tree = 0;(使用宏块树位元率控制会改善整体压缩率)

## x264线程

x264起多少个线程比较好 ？

建议线程数：

1、2、4、8

测试结论：

1、更多的线程会消耗更多总CPU时间片，因此在长期满载的机器上不宜使用多线程。

2、获得的时间收益随线程增多呈递减趋势，8线程以后尤为明显。

3、PNSR下降随线程数增加呈抛物递增趋势，16线程增加到24线程PSNR时下降了0.6之巨。

4、设置threads=auto时，线程数为逻辑CPU个数的1.5倍。

## x264各类型帧的大小及编码耗时

注：作参考，未必属实。

I帧、B帧、P帧都极大地受编码参数的影响。

通常情况下：

h264编码的帧由大到小依次为：

I > P > B

（互相之间约有5倍的差距）

x264的编码耗时由长到短依次为：

P > B > I

通常而言，较小的帧因为帧内压缩计算量(deblock、cabac等)小，所以耗时相对短。

P帧的编码耗时长是因为帧间压缩(宏块寻找、运动补偿等)耗时长所以提高了总体耗时。

另外：可以修改x264中的x264\_slices\_write函数来测量不同类型帧的编码耗时。

# ffmpeg编码参数和x264参数对照

下面表中涉及的参数直接在AVCodecContext结构中设置：

|  |  |  |  |
| --- | --- | --- | --- |
| **FFmpeg option** | | **x264 option** | |
| **Code** | **Console** | **Console** | **Code** |
| gop\_size | -g <frames> | –keyint | i\_keyint\_max |
| bit\_rate | -b <bits per second> | –bitrate | rc.i\_bitrate |
| **rc\_buffer\_size** | -bufsize <bits> | –vbv-bufsize | rc.i\_vbv\_buffer\_size |
| rc\_max\_rate | -maxrate <bits> | –vbv-maxrate | rc.i\_vbv\_max\_bitrate |
| max\_b\_frames | -bf <int> | –bframes | i\_bframe |
| keyint\_min | -keyint\_min <int> | –min-keyint | i\_keyint\_min |
| scenechange\_threshold | -sc\_threshold <int> | –scenecut | i\_scenecut\_threshold |
| qmin | -qmin <int> | –qpmin | rc.i\_qp\_min |
| qmax | -qmax <int> | –qpmax | rc.i\_qp\_max |
| max\_qdiff | -qdiff <int> | –qpstep | rc.i\_qp\_step |
| qcompress | -qcomp <float> | –qcomp | rc.f\_qcompress |
| qblur | -qblur <float> | –qblur | rc.f\_qblur |
| Refs | -refs <int> | –ref | i\_frame\_reference |
| me\_method | -me\_method <epzs,hex,umh,full> | –me | analyse.i\_me\_method |
| merange | -me\_range <int> | –merange | analyse.i\_me\_range |
| me\_subpel\_quality | -subq <int> | –subme | analyse.i\_subpel\_refine |
| trellis | -trellis <0,1,2> | –trellis | analyse.i\_trellis |
| noise\_reduction | -nr <int> | –nr | analyse.i\_noise\_reduction |
| level | -level <int> | –level | i\_level\_idc |
| bit\_rate\_tolerance | -bt <bits> | –ratetol = -bt / -b | rc.f\_rate\_tolerance |
| rc\_initial\_buffer\_occupancy | -rc\_init\_occupancy <bits> | –vbv-init = -rc\_init\_occupancy / -bufsize | rc.f\_vbv\_buffer\_init |
| b\_quant\_factor | -b\_qfactor <float> | –pbratio | rc.f\_pb\_factor |
| chromaoffset | -chromaoffset <int> | –chroma-qp-offset | analyse.i\_chroma\_qp\_offset |
| thread\_count | -threads <int> | –threads | i\_threads |
| me\_cmp | -cmp <-chroma/+chroma> | chroma-me | analyse.b\_chroma\_me |
| thread\_type | thread\_type | sliced\_threads | b\_sliced\_threads |

AVCodecContext结构中priv\_data可设置的参数见下表：

|  |
| --- |
| **priv\_data**(即**X264Context option**) |
| preset |
| tune |
| profile |
| fastfirstpass |
| level |
| stats |
| wpredp |
| x264opts |
| crf |
| crf\_max |
| cqp |
| aq\_mode |
| aq\_strength |
| psy |
| psy\_rd |
| rc\_lookahead |
| weightb |
| weightp |
| ssim |
| intra\_refresh |
| b\_bias |
| b\_pyramid |
| mixed\_refs |
| dct8x8 |
| fast\_pskip |
| aud |
| mbtree |
| deblock |
| cplxblur |
| partitions |
| direct\_pred |
| slice\_max\_size |
| stats |
| nal\_hrd |
| x264\_params |

其余x264参数设置，见下面格式，多个参数用冒号(:)隔开：

av\_opt\_set(AVCodecContext->priv\_data, "x264opts", "sync-lookahead=0: sliced-threads", 0);

# x264参数说明

注：下面说明不是最新版本，最新版本请参考x264 --fullhelp

预设

为了减少使用者花费时间和精力在命令列上而设计的一套系统。这些设定切换了什么选项可以从x264 --fullhelp的说明里得知。

profile

预设值：无

限制输出资料流的profile。如果指定了profile，它会覆写所有其他的设定。所以如果指定了profile，将会保证得到一个相容的资料流。如果设了此选项，将会无法使用无失真（lossless）编码（--[qp](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#qp) 0或--[crf](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#crf) 0）。

如果播放装置仅支援某个profile，则应该设此选项。大多数解码器都支援High profile，所以没有设定的必要。

可用的值：baseline, main, high

preset

预设值：medium

变更选项，以权衡压缩效率和编码速度。如果指定了预设，变更的选项将会在套用所有其他的参数之前套用。

通常应该将此设为所能忍受的最慢一个选项。

可用的值：ultrafast, superfast, veryfast, faster, fast, medium, slow, slower, veryslow, placebo

ultrafast

--no-8x8dct --aq-mode 0 --b-adapt 0

--bframes 0 --no-cabac --no-deblock

--no-mbtree --me dia --no-mixed-refs

--partitions none --rc-lookahead 0 --ref 1

--scenecut 0 --subme 0 --trellis 0

--no-weightb --weightp 0

Superfast

--no-mbtree --me dia --no-mixed-refs

--partitions i8x8,i4x4 --rc-lookahead 0 --ref 1

--subme 1 --trellis 0 --weightp 1

param->analyse.inter = X264\_ANALYSE\_I8x8|X264\_ANALYSE\_I4x4;

param->analyse.i\_me\_method = X264\_ME\_DIA;

param->analyse.i\_subpel\_refine = 1;

param->i\_frame\_reference = 1;

param->analyse.b\_mixed\_references = 0;

param->analyse.i\_trellis = 0;

param->rc.b\_mb\_tree = 0;

param->analyse.i\_weighted\_pred = X264\_WEIGHTP\_SIMPLE;

param->rc.i\_lookahead = 0;

tune

预设值：无

调整选项，以进一步最佳化为视讯的内容。如果指定了tune，变更的选项将会在--[preset](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#preset)之后，但所有其他的参数之前套用。

如果视讯的内容符合其中一个可用的调整值，则可以使用此选项，否则不要使用。

可用的值：film, animation, grain, stillimage, psnr, ssim, fastdecode, zerolatency

slow-firstpass

预设值：无

使用--[pass](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pass) 1会在解析命令列的最后套用以下设定：

--[ref](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#ref) 1

--[no-8x8dct](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#no-8x8dct)

--[partitions](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#partitions) i4x4 （如果最初有启用，否则为无）

--[me](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#me) dia

--[subme](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#subme) MIN(2, subme)

--[trellis](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#trellis) 0

可以使用--[slow-firstpass](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#slow-firstpass)来停用此功能。注意，使用--[preset](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#preset) placebo也会启用slow-firstpass。

参阅：--[pass](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pass)

帧类型选项

keyint

预设值：250

设定x264输出的资料流之最大IDR帧（亦称为关键帧）间隔。可以指定infinite让x264永远不要插入非场景变更的IDR帧。

IDR帧是资料流的“分隔符号”，所有帧都无法从IDR帧的另一边参照资料。因此，IDR帧也是I帧，所以它们不从任何其他帧参照资料。这意味着它们可以用作视讯的搜寻点（seek points）。

注意，I帧通常明显大于P/B帧（在低动态场景通常为10倍大或更多），所以当它们与极低的VBV设定合并使用时会打乱位元率控制。在这些情况下，研究--[intra-refresh](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#intra-refresh)。

预设值对于大多数视讯没啥问题。在为蓝光、广播、即时资料流或某些其他特殊情况编码时，可能需要更小的GOP长度（通常等于帧率）。

参阅：--[min-keyint](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#min-keyint), --[scenecut](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#scenecut), --[intra-refresh](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#intra-refresh)

min-keyint

预设值：自动 （MIN(--[keyint](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#keyint) / 10, --[fps](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#fps))）

设定IDR帧之间的最小长度。

IDR帧的说明可以参阅--[keyint](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#keyint)。过小的keyint范围会导致“不正确的”IDR帧位置（例如闪屏场景）。此选项限制在每个IDR帧之后，要有多少帧才可以再有另一个IDR帧的最小长度。

min-keyint的最大允许值是--[keyint](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#keyint)/2+1。

建议：预设值，或者等于帧率

参阅：--[keyint](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#keyint), --[scenecut](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#scenecut)

no-scenecut

预设值：无

完全停用弹性I帧决策（adaptive I-frame decision）。

参阅：--[scenecut](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#scenecut)

scenecut

预设值：40

设定I/IDR帧位置的阈值（场景变更侦测）。

x264为每一帧计算一个度量值，来估计与前一帧的不同程度。如果该值低于scenecut，则算侦测到一个“场景变更”。如果此时与最近一个IDR帧的距离低于--[min-keyint](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#min-keyint)，则放置一个I帧，否则放置一个IDR帧。越大的scenecut值会增加侦测到场景变更的数目。场景变更是如何比较的详细资讯可以参阅<http://forum.doom9.org/showthread.php?t=121116>。

将scenecut设为0相当于设定--[no-scenecut](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#no-scenecut)。

建议：预设值

参阅：--[keyint](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#keyint), --[min-keyint](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#min-keyint), --[no-scenecut](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#no-scenecut)

intra-refresh

预设值：无

停用IDR帧，作为替代x264会为每隔--[keyint](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#keyint)的帧的每个巨集区块（macroblock）使用内部编码（intra coding）。区块是以一个水平卷动的行刷新，称为刷新波（refresh wave）。这有利于低延迟的资料流，使它有可能比标准的IDR帧达到更加固定的帧大小。它也增强了视讯资料流对封包遗失的恢复能力。此选项会降低压缩效率，因此必要时才使用。

有趣的事：

第一帧仍然是IDR帧。

内部区块（Intra-blocks）仅处于P帧里，刷新波在一或多个B帧后的第一个P帧更广泛。

压缩效率的损失主要来自于在刷新波上左侧（新）的巨集区块无法参照右侧（旧）的资料。

bframes

预设值：3

设定x264可以使用的最大并行B帧数。

没有B帧时，一个典型的x264资料流有着像这样的帧类型：IPPPPP...PI。当设了--bframes 2时，最多两个连续的P帧可以被B帧取代，就像：IBPBBPBPPPB...PI。

B帧类似于P帧，除了B帧还能从它之后的帧做动态预测（motion prediction）。就压缩比来说效率会大幅提高。它们的平均品质是由--[pbratio](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pbratio)所控制。

有趣的事：

x264还区分两种不同种类的B帧。"B"是代表一个被其他帧作为参照帧的B帧（参阅--[b-pyramid](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#b-pyramid)），而"b"则代表一个不被其他帧作为参照帧的B帧。如果看到一段混合的"B"和"b"，原因通常与上述有关。当差别并不重要时，通常就以"B"代表所有B帧。

x264是如何为每个候选帧选定为P帧或B帧的详细资讯可以参阅<http://article.gmane.org/gmane.comp.video.ffmpeg.devel/29064>。在此情况下，帧类型看起来会像这样（假设--bframes 3）：IBBBPBBBPBPI。

参阅：--[b-bias](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#b-bias), --[b-pyramid](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#b-pyramid), --[ref](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#ref), --[pbratio](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pbratio), --[partitions](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#partitions), --[weightb](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#weightb)

b-adapt

预设值：1

设定弹性B帧位置决策演算法。此设定控制x264如何决定要放置P帧或B帧。

0：停用，总是挑选B帧。这与旧的no-b-adapt设定相同作用。

1：“快速”演算法，较快，越大的--[bframes](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bframes)值会稍微提高速度。当使用此模式时，基本上建议搭配--[bframes](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bframes) 16使用。

2：“最佳”演算法，较慢，越大的--[bframes](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bframes)值会大幅降低速度。

注意：对于多重阶段（multi-pass）编码，仅在第一阶段（first pass）才需要此选项，因为帧类型在此时已经决定完了。

b-bias

预设值：0

控制使用B帧而不使用P帧的可能性。大于0的值增加偏向B帧的加权，而小于0的值则相反。范围是从-100到100。100并不保证全是B帧（要全是B帧该使用--[b-adapt](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#b-adapt) 0），而-100也不保证全是P帧。

仅在你认为能比x264做出更好的位元率控制决策时才使用此选项。

参阅：--[bframes](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bframes), --[ipratio](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#ipratio)

b-pyramid

预设值：normal

允许B帧作为其他帧的参照帧。没有此设定时，帧只能参照I/P帧。虽然I/P帧因其较高的品质作为参照帧更有价值，但B帧也是很有用的。作为参照帧的B帧会得到一个介于P帧和普通B帧之间的量化值。b-pyramid需要至少两个以上的--[bframes](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bframes)才会运作。

如果是在为蓝光编码，须使用none或strict。

none：不允许B帧作为参照帧。

strict：每minigop允许一个B帧作为参照帧，这是蓝光标准强制执行的限制。

normal：每minigop允许多个B帧作为参照帧。

参阅：--[bframes](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bframes), --[refs](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#refs), --[no-mixed-refs](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#no-mixed-refs)

open-gop

预设值：none

open-gop是一个提高效率的编码技术。有三种模式：

none：停用open-gop。

normal：启用open-gop。

bluray：启用open-gop。一个效率较低的open-gop版本，因为normal模式无法用于蓝光编码。

某些解码器不完全支援open-gop资料流，这就是为什么此选项并未预设为启用。如果想启用open-gop，应该先测试所有可能用来拨放的解码器。

open-gop的说明可以参阅<http://forum.doom9.org/showthread.php?p=1300124#post1300124>。

no-cabac

预设值：无

停用弹性内容的二进位算数编码（CABAC：Context Adaptive Binary Arithmetic Coder）资料流压缩，切换回效率较低的弹性内容的可变长度编码（CAVLC：Context Adaptive Variable Length Coder）系统。大幅降低压缩效率（通常10~20%）和解码的硬体需求。

ref

预设值：3

控制解码图片缓冲（DPB：Decoded Picture Buffer）的大小。范围是从0到16。总之，此值是每个P帧可以使用先前多少帧作为参照帧的数目（B帧可以使用的数目要少一或两个，取决于它们是否作为参照帧）。可以被参照的最小ref数是1。

还要注意的是，H.264规格限制了每个level的DPB大小。如果遵守[Level 4.1](http://en.wikipedia.org/wiki/H.264/MPEG-4_AVC#Levels)规格，720p和1080p视讯的最大ref数分别是9和4。

参阅：--[b-pyramid](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#b-pyramid), --[no-mixed-refs](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#no-mixed-refs), --[level](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#level)

no-deblock

预设值：无

完全停用循环筛选（loop filter）。不建议。

参阅：--[deblock](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#deblock)

deblock

预设值：0:0

控制循环筛选（亦称为持续循环去区块(inloop deblocker)），这是H.264标准的一部分。就性价比来说非常有效率。

可以在<http://forum.doom9.org/showthread.php?t=109747>找到loop滤镜的参数是如何运作的说明（参阅第一个帖子和akupenguin的回复）。

参阅：--[no-deblock](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#no-deblock)

slices

预设值：无

设定每帧的切片数，而且强制为矩形切片（会被--[slice-max-size](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#slice-max-size)或--[slice-max-mbs](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#slice-max-mbs)覆写）。

如果是在为蓝光编码，将值设为4。否则，不要使用此选项，除非你知道真的有必要。

参阅：--[slice-max-size](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#slice-max-size), --[slice-max-mbs](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#slice-max-mbs)

slice-max-size

预设值：无

设定最大的切片大小（单位是位元组），包括估计的NAL额外负荷（overhead）。（目前与--[interlaced](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#interlaced)不相容）

参阅：--[slices](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#slices)

slice-max-mbs

预设值：无

设定最大的切片大小（单位是巨集区块）。（目前与--[interlaced](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#interlaced)不相容）

参阅：--[slices](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#slices)

tff

预设值：无

启用交错式编码并指定顶场优先（top field first）。x264的交错式编码使用MBAFF，本身效率比渐进式编码差。出于此原因，仅在打算于交错式显示器上播放视讯时，才应该编码为交错式（或者视讯在送给x264之前无法进行去交错）。此选项会自动启用--[pic-struct](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pic-struct)。

bff

预设值：无

启用交错式编码并指定底场优先（bottom field first）。详细资讯可以参阅--[tff](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#tff)。

constrained-intra

预设值：无

启用限制的内部预测（constrained intra prediction），这是SVC编码的基础层（base layer）所需要的。既然EveryoneTM忽略SVC，你同样可以忽略此选项。

pulldown

预设值：none

使用其中一个预设模式将渐进式、固定帧率的输入资料流标志上软胶卷过带（soft telecine）。软胶卷过带在<http://trac.handbrake.fr/wiki/Telecine>有更详细的解释。

可用的预设：none, 22, 32, 64, double, triple, euro

指定除了none以外的任一模式会自动启用--[pic-struct](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pic-struct)。

fake-interlaced

预设值：无

将资料流标记为交错式，即使它并未以交错式来编码。用于编码25p和30p为符合蓝光标准的视讯。

frame-packing

预设值：无

如果在编码3D视讯，此参数设定一个位元资料流（bitstream）旗标，用来告诉解码器3D视讯是如何被封装。相关的值和它们的意义可以从x264 --fullhelp的说明里得知。

位元率控制

qp

预设值：无

三种位元率控制方法之一。设定x264以固定量化值（Constant Quantizer）模式来编码视讯。这里给的值是指定P帧的量化值。I帧和B帧的量化值则是从--[ipratio](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#ipratio)和--[pbratio](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pbratio)中取得。CQ模式把某个量化值作为目标，这意味着最终档案大小是未知的（虽然可以透过一些方法来准确地估计）。将值设为0会产生无失真输出。对于相同视觉品质，qp会比--[crf](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#crf)产生更大的档案。qp模式也会停用弹性量化，因为按照定义“固定量化值”意味着没有弹性量化。

此选项与--[bitrate](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bitrate)和--[crf](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#crf)互斥。各种位元率控制系统的详细资讯可以参阅<http://git.videolan.org/?p=x264.git;a=blob_plain;f=doc/ratecontrol.txt;hb=HEAD>。

虽然qp不需要lookahead来执行因此速度较快，但通常应该改用--[crf](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#crf)。

参阅：--[bitrate](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bitrate), --[crf](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#crf), --[ipratio](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#ipratio), --[pbratio](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pbratio)

bitrate

预设值：无

三种位元率控制方法之二。以目标位元率模式来编码视讯。目标位元率模式意味着最终档案大小是已知的，但最终品质则未知。x264会尝试把给定的位元率作为整体平均值来编码视讯。参数的单位是千位元/秒（8位元=1位元组）。注意，1千位元(kilobit)是1000位元，而不是1024位元。

此设定通常与--[pass](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pass)在两阶段（two-pass）编码一起使用。

此选项与--[qp](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#qp)和--[crf](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#crf)互斥。各种位元率控制系统的详细资讯可以参阅<http://git.videolan.org/?p=x264.git;a=blob_plain;f=doc/ratecontrol.txt;hb=HEAD>。

参阅：--[qp](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#qp), --[crf](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#crf), --[ratetol](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#ratetol), --[pass](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pass), --[stats](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#stats)

crf

预设值：23.0

最后一种位元率控制方法：固定位元率系数（Constant Ratefactor）。当qp是把某个量化值作为目标，而bitrate是把某个档案大小作为目标时，crf则是把某个“品质”作为目标。构想是让crf n提供的视觉品质与qp n相同，只是档案更小一点。crf值的度量单位是“位元率系数（ratefactor）”。

CRF是借由降低“较不重要”的帧之品质来达到此目的。在此情况下，“较不重要”是指在复杂或高动态场景的帧，其品质不是很耗费位元数就是不易察觉，所以会提高它们的量化值。从这些帧里所节省下来的位元数被重新分配到可以更有效利用的帧。

CRF花费的时间会比两阶段编码少，因为两阶段编码中的“第一阶段”被略过了。另一方面，要预测CRF编码的最终位元率是不可能的。根据情况哪种位元率控制模式更好是由你来决定。

此选项与--[qp](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#qp)和--[bitrate](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bitrate)互斥。各种位元率控制系统的详细资讯可以参阅<http://git.videolan.org/?p=x264.git;a=blob_plain;f=doc/ratecontrol.txt;hb=HEAD>。

参阅：--[qp](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#qp), --[bitrate](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bitrate)

rc-lookahead

预设值：40

设定mb-tree位元率控制和vbv-lookahead使用的帧数。最大允许值是250。

对于mb-tree部分，增加帧数带来更好的效果但也会更慢。mb-tree使用的最大缓冲值是MIN(rc-lookahead, --[keyint](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#keyint))。

对于vbv-lookahead部分，当使用vbv时，增加帧数带来更好的稳定性和准确度。vbv-lookahead使用的最大值是：

MIN(rc-lookahead, MAX(--[keyint](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#keyint), MAX(--[vbv-maxrate](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#vbv-maxrate), --[bitrate](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bitrate)) / --[vbv-bufsize](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#vbv-bufsize) \* --[fps](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#fps)))

参阅：--[no-mbtree](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#no-mbtree), --[vbv-bufsize](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#vbv-bufsize), --[vbv-maxrate](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#vbv-maxrate)

什么是Macroblock Tree  
Macroblock Tree是一个基于macroblock的qp控制方法。MB Tree的工作原理类似于古典的qp compression，只不过qcomp处理的对象是整张frame而MB Tree针对的是每个MB进行处理。工作过程简单来说，是对于每个MB，向前预测一定数量的帧（该数量由rc-lookahead和keyint的较小值决定）中该MB被参考的情况，根据引用次数的多寡，决定对该MB使用何种大小的qp进行quantization。而qp的大小与被参考次数成反比，也就是说，对于被参考次数多的MB，264的解码器认为此对应于缓慢变化的场景，因此给与比较高的质量（比较低的qp数值）。至于视频的变化率与人眼感知能力的关系，这是一个基于主观测试的经验结果：视频变化率越大 人眼的敏感度越低，也就是说，人眼可以容忍快速变化场景的某些缺陷，但相对而言某些平滑场景的缺陷，人眼则相当敏感。注意此处说的平滑，指的是沿时间维度上场景的变化频率，而非普通意义上的像素域中的场景。

vbv-maxrate

预设值：0

设定重新填满VBV缓冲的最大位元率。

VBV会降低品质，所以必要时才使用。

参阅：--[vbv-bufsize](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#vbv-bufsize), --[vbv-init](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#vbv-init), [VBV编码建议](http://www.nmm-hd.org/doc/index.php?title=X264%E7%B7%A8%E7%A2%BC%E5%BB%BA%E8%AD%B0&action=edit&redlink=1)

vbv-bufsize

预设值：0

设定VBV缓冲的大小（单位是千位元）。

VBV会降低品质，所以必要时才使用。

参阅：--[vbv-maxsize](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#vbv-maxsize), --[vbv-init](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#vbv-init), [VBV编码建议](http://www.nmm-hd.org/doc/index.php?title=X264%E7%B7%A8%E7%A2%BC%E5%BB%BA%E8%AD%B0&action=edit&redlink=1)

vbv-init

预设值：0.9

设定VBV缓冲必须填满多少才会开始播放。

如果值小于1，初始的填满量是：vbv-init \* vbv-bufsize。否则该值即是初始的填满量（单位是千位元）。

参阅：--[vbv-maxsize](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#vbv-maxsize), --[vbv-bufsize](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#vbv-bufsize), [VBV编码建议](http://www.nmm-hd.org/doc/index.php?title=X264%E7%B7%A8%E7%A2%BC%E5%BB%BA%E8%AD%B0&action=edit&redlink=1)

crf-max

预设值：无

一个类似--[qpmax](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#qpmax)的设定，除了指定的是最大位元率系数而非最大量化值。当使用--[crf](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#crf)且启用VBV时，此选项才会运作。它阻止x264降低位元率系数（亦称为“品质”）到低于给定的值，即使这样做会违反VBV的条件约束。此设定主要适用于自订资料流伺服器。详细资讯可以参阅<http://git.videolan.org/gitweb.cgi/x264.git/?a=commit;h=81eee062a4ce9aae1eceb3befcae855c25e5ec52>。

参阅：--[crf](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#crf), --[vbv-maxrate](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#vbv-maxrate), --[vbv-bufsize](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#vbv-bufsize)

qpmin

预设值：0

定义x264可以使用的最小量化值。量化值越小，输出就越接近输入。到了一定的值，x264的输出看起来会跟输入一样，即使它并不完全相同。通常没有理由允许x264花费比这更多的位元数在任何特定的巨集区块上。

当弹性量化启用时（预设启用），不建议提高qpmin，因为这会降低帧里面平面背景区域的品质。

参阅：--[qpmax](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#qpmax), --[ipratio](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#ipratio)

关于qpmin的预设值：在x264 r1795版本之前，该选项预设值为10。

qpmax

预设值：51

定义x264可以使用的最大量化值。预设值51是H.264规格可供使用的最大量化值，而且品质极低。此预设值有效地停用了qpmax。如果想要限制x264可以输出的最低品质，可以将此值设小一点（通常30~40），但通常并不建议调整此值。

参阅：--[qpmin](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#qpmin), --[pbratio](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pbratio), --[crf-max](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#crf-max)

qpstep

预设值：4

设定两帧之间量化值的最大变更幅度。

ratetol

预设值：1.0

此参数有两个目的：

在一阶段位元率编码时，此设定控制x264可以偏离目标平均位元率的百分比。可以指定inf来完全停用溢出侦测（overflow detection）。可以设定的最小值是0.01。值设得越大，x264可以对接近电影结尾的复杂场景作出越好的反应。此目的的度量单位是百分比（例如，1.0等于允许1%的位元率偏差）。

很多电影（例如动作片）在电影结尾时是最复杂的。因为一阶段编码并不知道这一点，结尾所需的位元数通常被低估。将ratetol设为inf可以减轻此情况，借由允许编码以更像--[crf](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#crf)的模式运行，但档案大小会暴增。

当VBV启用时（即指定了--vbv-开头的选项），此设定也会影响VBV的强度。值设得越大，允许VBV在冒着可能违反VBV设定的风险下有越大的波动。

ipratio

预设值：1.40

修改I帧量化值相比P帧量化值的目标平均增量。越大的值会提高I帧的品质。

参阅：--[pbratio](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pbratio)

pbratio

预设值：1.30

修改B帧量化值相比P帧量化值的目标平均减量。越大的值会降低B帧的品质。当mbtree启用时（预设启用），此设定无作用，mbtree会自动计算最佳值。

参阅：--[ipratio](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#ipratio)

chroma-qp-offset

预设值：0

在编码时增加色度平面量化值的偏移。偏移可以为负数。

当使用psy-rd或psy-trellis时，x264自动降低此值来提高亮度的品质，其后降低色度的品质。这些设定的预设值会使chroma-qp-offset再减去2。

注意：x264仅在同一量化值编码亮度平面和色度平面，直到量化值29。在此之后，色度逐步以比亮度低的量被量化，直到亮度在q51和色度在q39为止。此行为是由H.264标准所要求。

aq-mode

预设值：1

弹性量化模式。没有AQ时，x264很容易分配不足的位元数到细节较少的部分。AQ是用来更好地分配视讯里所有巨集区块之间的可用位元数。此设定变更AQ会重新分配位元数到什么范围里：

0：完全不使用AQ。

1：允许AQ重新分配位元数到整个视讯和帧内。

2：自动变化（Auto-variance）AQ，会尝试对每帧调整强度。（实验性的）

参阅：--[aq-strength](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#aq-strength)

aq-strength

预设值：1.0

弹性量化强度。设定AQ偏向低细节（平面）的巨集区块之强度。不允许为负数。0.0~2.0以外的值不建议。

参阅：--[aq-mode](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#aq-mode)

pass

预设值：无

此为两阶段编码的一个重要设定。它控制x264如何处理--[stats](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#stats)档案。有三种设定：

1：建立一个新的统计资料档案。在第一阶段使用此选项。

2：读取统计资料档案。在最终阶段使用此选项。

3：读取统计资料档案并更新。

统计资料档案包含每个输入帧的资讯，可以输入到x264以改善输出。构想是执行第一阶段来产生统计资料档案，然后第二阶段将建立一个最佳化的视讯编码。改善的地方主要是从更好的位元率控制中获益。

参阅：--[stats](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#stats), --[bitrate](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bitrate), --[slow-firstpass](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#slow-firstpass), [X264统计资料档案](http://www.nmm-hd.org/doc/index.php?title=X264%E7%B5%B1%E8%A8%88%E8%B3%87%E6%96%99%E6%AA%94%E6%A1%88&action=edit&redlink=1)

stats

预设值："x264\_2pass.log"

设定x264读取和写入统计资料档案的位置。

参阅：--[pass](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pass), [X264统计资料档案](http://www.nmm-hd.org/doc/index.php?title=X264%E7%B5%B1%E8%A8%88%E8%B3%87%E6%96%99%E6%AA%94%E6%A1%88&action=edit&redlink=1)

no-mbtree

预设值：无

停用巨集区块树（macroblock tree）位元率控制。使用巨集区块树位元率控制会改善整体压缩率，借由追踪跨帧的时间传播（temporal propagation）并相应地加权。除了已经存在的统计资料档案之外，多重阶段编码还需要一个新的统计资料档案。

建议：预设值

参阅：--[rc-lookahead](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#rc-lookahead)

qcomp

预设值：0.60

量化值曲线压缩系数。0.0是固定位元率，1.0则是固定量化值。

当mbtree启用时，它会影响mbtree的强度（qcomp越大，mbtree越弱）。

建议：预设值

参阅：--[cplxblur](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#cplxblur), --[qblur](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#qblur)

cplxblur

预设值：20.0

以给定的半径范围套用高斯模糊（gaussian blur）于量化值曲线。这意味着分配给每个帧的量化值会被它的邻近帧模糊掉，以此来限制量化值波动。

参阅：--[qcomp](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#qcomp), --[qblur](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#qblur)

qblur

预设值：0.5

在曲线压缩之后，以给定的半径范围套用高斯模糊于量化值曲线。不怎么重要的设定。

参阅：--[qcomp](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#qcomp), --[cplxblur](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#cplxblur)

zones

预设值：无

调整视讯的特定片段之设定。可以修改每区段的大多数x264选项。

一个单一区段的形式为<起始帧>,<结束帧>,<选项>。

多个区段彼此以"/"分隔。

选项：

这两个是特殊选项。每区段只能设定其中一个，而且如果有设定其中一个，它必须为该区段列出的第一个选项：

b=<浮点数> 套用位元率乘数在此区段。在额外调整高动态和低动态场景时很有用。

q=<整数> 套用固定量化值在此区段。在套用于一段范围的帧时很有用。

其他可用的选项如下：

ref=<整数>

b-bias=<整数>

scenecut=<整数>

no-deblock

deblock=<整数>:<整数>

deadzone-intra=<整数>

deadzone-inter=<整数>

direct=<字串>

merange=<整数>

nr=<整数>

subme=<整数>

trellis=<整数>

(no-)chroma-me

(no-)dct-decimate

(no-)fast-pskip

(no-)mixed-refs

psy-rd=<浮点数>:<浮点数>

me=<字串>

no-8x8dct

b-pyramid=<字串>

限制：

一个区段的参照帧数无法超过--[ref](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#ref)所指定的大小。

无法开启或关闭scenecut；如果--[scenecut](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#scenecut)最初为开启（>0），则只能改变scenecut的大小。

如果使用--[me](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#me) esa/tesa，merange无法超过最初所指定的大小。

如果--[subme](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#subme)最初指定为0，则无法变更subme。

如果--[me](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#me)最初指定为dia、hex或umh，则无法将me设为esa为tesa。

范例：0,1000,b=2/1001,2000,q=20,me=3,b-bias=-1000

建议：预设值

qpfile

预设值：无

手动覆写标准的位元率控制。指定一个档案，为指定的帧赋予量化值和帧类型。格式为“帧号 帧类型 量化值”。例如：

0 I 18 < IDR (key) I-frame  
1 P 18 < P-frame  
2 B 18 < Referenced B-frame  
3 i 18 < Non-IDR (non-key) I-frame  
4 b 18 < Non-referenced B-frame  
5 K 18 < Keyframe\*

不需要指定每个帧。

使用-1作为所需的量化值允许x264自行选择最佳的量化值，在只需设定帧类型时很有用。

在指定了大量的帧类型和量化值时仍然让x264间歇地自行选择，会降低x264的效能。

"Keyframe"是一个泛用关键帧／搜寻点，如果--[open-gop](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#open-gop)是none则等同于一个IDR I帧，否则等同于一个加上Recovery Point SEI旗标的Non-IDR I帧。

分析

partitions

预设值：p8x8,b8x8,i8x8,i4x4

H.264视讯在压缩过程中划分为16x16的巨集区块。这些区块可以进一步划分为更小的分割，这就是此选项要控制的部分。

此选项可以启用个别分割。分割依不同帧类型启用。

可用的分割：p8x8, p4x4, b8x8, i8x8, i4x4, none, all

I：i8x8、i4x4。

P：p8x8（也会启用p16x8/p8x16）、p4x4（也会启用p8x4/p4x8）。

B：b8x8（也会启用b16x8/b8x16）。

p4x4通常不怎么有用，而且性价比极低。

参阅：--[no-8x8dct](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#no-8x8dct)

direct

预设值：spatial

设定"direct"动态向量（motion vectors）的预测模式。有两种模式可用：spatial和temporal。可以指定none来停用direct动态向量，和指定auto来允许x264在两者之间切换为适合的模式。如果设为auto，x264会在编码结束时输出使用情况的资讯。auto最适合用于两阶段编码，但也可用于一阶段编码。在第一阶段auto模式，x264持续记录每个方法执行到目前为止的好坏，并从该记录挑选下一个预测模式。注意，仅在第一阶段有指定auto时，才应该在第二阶段指定auto；如果第一阶段不是指定auto，第二阶段将会预设为temporal。none模式会浪费位元数，因此强烈不建议。

建议：auto

no-weightb

预设值：无

H.264允许“加权”B帧的参照，它允许变更每个参照影响预测图片的程度。此选项停用该功能。

建议：预设值

weightp

预设值：2

使x264能够使用明确加权预测（explicit weighted prediction）来改善P帧的压缩。亦改善淡入／淡出的品质。模式越高越慢。

注意：在为Adobe Flash编码时，将值设为1，否则它的解码器会产生不自然痕迹（artifacts）。Flash 10.1修正了此bug。

模式：

0：停用。

1：简易：分析淡入／淡出，但不分析重复参照帧。

2：智慧：分析淡入／淡出和重复参照帧。

me

预设值：hex

设定全像素（full-pixel）动态估算（motion estimation）的方法。有五个选项：

dia（diamond）：最简单的搜寻方法，起始于最佳预测器（predictor），检查上、左、下、右方一个像素的动态向量，挑选其中最好的一个，并重复此过程直到它不再找到任何更好的动态向量为止。

hex（hexagon）：由类似策略组成，除了它使用周围6点范围为2的搜寻，因此叫做六边形。它比dia更有效率且几乎没有变慢，因此作为一般用途的编码是个不错的选择。

umh（uneven multi-hex）：比hex更慢，但搜寻复杂的多六边形图样以避免遗漏难以找到的动态向量。不像hex和dia，merange参数直接控制umh的搜寻半径，允许增加或减少广域搜寻的大小。

esa（exhaustive）：一种在merange内整个动态搜寻空间的高度最佳化智慧搜寻。虽然速度较快，但数学上相当于搜寻该区域每个单一动态向量的暴力（bruteforce）方法。不过，它仍然比UMH还要慢，而且没有带来很大的好处，所以对于日常的编码不是特别有用。

tesa（transformed exhaustive）：一种尝试接近在每个动态向量执行Hadamard转换法比较的效果之演算法，就像exhaustive，但效果好一点而速度慢一点。

参阅：--[merange](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#merange)

merange

预设值：16

merange控制动态搜寻的最大范围（单位是像素）。对于hex和dia，范围限制在4~16。对于umh和esa，它可以增加到超过预设值16来允许范围更广的动态搜寻，对于HD视讯和高动态镜头很有用。注意，对于umh、esa和tesa，增加merange会大幅减慢编码速度。

参阅：--[me](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#me)

mvrange

预设值：-1 （自动）

设定动态向量的最大（垂直）范围（单位是像素）。预设值依level不同：

Level 1/1b：64。

Level 1.1~2.0：128。

Level 2.1~3.0：256。

Level 3.1+：512。

注意：如果想要手动覆写mvrange，在设定时从上述值减去0.25（例如--mvrange 127.75）。

建议：预设值

mvrange-thread

预设值：-1 （自动）

设定执行绪之间的最小动态向量缓冲。不要碰它。

建议：预设值

subme

预设值：7

设定子像素（subpixel）估算复杂度。值越高越好。层级1~5只是控制子像素细分（refinement）强度。层级6为模式决策启用RDO，而层级8为动态向量和内部预测模式启用RDO。RDO层级明显慢于先前的层级。

使用小于2的值不但会启用较快且品质较低的lookahead模式，而且导致较差的--[scenecut](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#scenecut)决策，因此不建议。

可用的值：

0：Fullpel only

1：QPel SAD 1 iteration

2：QPel SATD 2 iterations

3：HPel on MB then QPel

4：Always QPel

5：Multi QPel + bi-directional motion estimation

6：RD on I/P frames

7：RD on all frames

8：RD refinement on I/P frames

9：RD refinement on all frames

10：QP-RD (requires --[trellis](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#trellis)=2, --[aq-mode](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#aq-mode)>0)

建议：预设值，或者更高，除非速度非常重要

psy-rd

预设值：1.0:0.0

第一个数是Psy-RDO的强度（需要subme>=6）。第二个数是Psy-Trellis的强度（需要trellis>=1）。注意，Trellis仍然被视为“实验性的”，而且几乎可以肯定至少卡通不适合使用。

psy-rd的解释可以参阅<http://forum.doom9.org/showthread.php?t=138293>。

no-psy

预设值：无

停用所有会降低PSNR或SSIM的视觉最佳化。这也会停用一些无法透过x264的命令列引数设定的内部psy最佳化。

建议：预设值

no-mixed-refs

预设值：无

混合参照会以每个8x8分割为基础来选取参照，而不是以每个巨集区块为基础。当使用多个参照帧时这会改善品质，虽然要损失一些速度。设定此选项会停用该功能。

建议：预设值

参阅：--[ref](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#ref)

no-chroma-me

预设值：无

通常，亮度（luma）和色度（chroma）两个平面都会做动态估算。此选项停用色度动态估算来提高些微速度。

建议：预设值

no-8x8dct

预设值：无

弹性8x8离散余弦转换（Adaptive 8x8 DCT）使x264能够智慧弹性地使用I帧的8x8转换。此选项停用该功能。

建议：预设值

trellis

预设值：1

执行[Trellis quantization](http://en.wikipedia.org/wiki/Trellis_quantization)来提高效率。

0：停用。

1：只在一个巨集区块的最终编码上启用。

2：在所有模式决策上启用。

在巨集区块时提供了速度和效率之间的良好平衡。在所有决策时则更加降低速度。

建议：预设值

注意：需要--[cabac](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#no-cabac)

no-fast-pskip

预设值：无

停用P帧的早期略过侦测（early skip detection）。非常轻微地提高品质，但要损失很多速度。

建议：预设值

no-dct-decimate

预设值：无

DCT Decimation会舍弃它认为“不必要的”DCT区块。这会改善编码效率，而降低的品质通常微不足道。设定此选项会停用该功能。

建议：预设值

nr

预设值：0

执行快速的杂讯削减（noise reduction）。根据此值估算影片的杂讯，并借由在量化之前舍弃小细节来尝试移除杂讯。这可能比不上优良的外部杂讯削减筛选的品质，但它执行得非常快。

建议：预设值，或者100~1000

deadzone-inter/intra

预设值：21/11

设定inter/intra亮度量化反应区（deadzone）的大小。反应区的范围应该在0~32。此值设定x264会任意舍弃而不尝试保留细微细节的层级。非常细微的细节既难以看见又耗费位元数，舍弃这些细节可以不用浪费位元数在视讯的此类低收益画面上。反应区与--[trellis](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#trellis)[不相容](http://forum.doom9.org/showthread.php?p=1031142#post1031142)。

建议：预设值

cqm

预设值：flat

设定所有自订量化矩阵（custom quantization matrices）为内建的预设之一。内建预设有flat和JVT。

建议：预设值

参阅：--[cqmfile](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#cqmfile)

cqmfile

预设值：无

从一个指定的JM相容档案来设定自订量化矩阵。覆写所有其他--cqm开头的选项。

建议：预设值

参阅：--[cqm](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#cqm)

cqm4\* / cqm8\*

预设值：无

--cqm4：设定所有4x4量化矩阵。需要16个以逗号分隔的整数清单。

--cqm8：设定所有8x8量化矩阵。需要64个以逗号分隔的整数清单。

--cqm4i、--cqm4p、--cqm8i、--cqm8p：设定亮度和色度量化矩阵。

--cqm4iy、--cqm4ic、--cqm4py、--cqm4pc：设定个别量化矩阵。

建议：预设值

视讯可用性资讯

这些选项在输出资料流里设定一个旗标，旗标可以被解码器读取并采取可能的动作。值得一提的是大多数选项在大多数情况下毫无意义，而且通常被解码器忽略。

overscan

预设值：undef

如何处理溢出扫描（overscan）。溢出扫描的意思是装置只显示影像的一部分。

可用的值：

undef：未定义。

show：指示要显示整个影像。理论上如果有设定则必须被遵守。

crop：指示此影像适合在有溢出扫描功能的装置上播放。不一定被遵守。

建议：在编码之前裁剪（Crop），然后如果装置支援则使用show，否则不理会

videoformat

预设值：undef

指示此视讯在编码／数位化（digitizing）之前是什么格式。

可用的值：component, pal, ntsc, secam, mac, undef

建议：来源视讯的格式，或者未定义

fullrange

预设值：off

指示是否使用亮度和色度层级的全范围。如果设为off，则会使用有限范围。

详细资讯可以参阅<http://en.wikipedia.org/wiki/YCbCr>。

建议：如果来源是从类比视讯数位化，将此设为off。否则设为on

colorprim

预设值：undef

设定以什么色彩原色转换成RGB。

可用的值：undef, bt709, bt470m, bt470bg, smpte170m, smpte240m, film

详细资讯可以参阅<http://en.wikipedia.org/wiki/RGB_color_space>和<http://en.wikipedia.org/wiki/YCbCr>。

建议：预设值，除非你知道来源使用什么色彩原色

transfer

预设值：undef

设定要使用的光电子（opto-electronic）传输特性（设定用于修正的色差补正(gamma)曲线）。

可用的值：undef, bt709, bt470m, bt470bg, linear, log100, log316, smpte170m, smpte240m

详细资讯可以参阅<http://en.wikipedia.org/wiki/Gamma_correction>。

建议：预设值，除非你知道来源使用什么传输特性

colormatrix

预设值：undef

设定用于从RGB原色中取得亮度和色度的矩阵系数。

可用的值：undef, bt709, fcc, bt470bg, smpte170m, smpte240m, GBR, YCgCo

详细资讯可以参阅<http://en.wikipedia.org/wiki/YCbCr>。

建议：来源使用的矩阵，或者预设值

chromaloc

预设值：0

设定色度采样位置（如[ITU-T规格](http://www.itu.int/rec/T-REC-H.264/en)的附录E所定义）

可用的值：0~5

参阅[x264的vui.txt](http://git.videolan.org/?p=x264.git;a=blob;f=doc/vui.txt)。

建议：

如果是从正确次采样4:2:0的MPEG1转码，而且没有做任何色彩空间转换，则应该将此选项设为1。

如果是从正确次采样4:2:0的MPEG2转码，而且没有做任何色彩空间转换，则应该将此选项设为0。

如果是从正确次采样4:2:0的MPEG4转码，而且没有做任何色彩空间转换，则应该将此选项设为0。

否则，维持预设值。

nal-hrd

预设值：none

标志HRD资讯。这是蓝光资料流、电视广播和几个其他专业范围所需要的。

可用的值：

none：不指定HRD资讯。

vbr：指定HRD资讯。

cbr：指定HRD资讯并以--[bitrate](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bitrate)指定的位元率来封装位元资料流。需要--[bitrate](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bitrate)模式的位元率控制。

建议：预设值，除非需要标志此资讯

参阅：--[vbv-bufsize](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#vbv-bufsize), --[vbv-maxrate](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#vbv-maxrate), --[aud](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#aud)

pic-struct

预设值：无

强制在Picture Timing SEI里传送pic\_struct。

当使用--[pulldown](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pulldown)或--[tff](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#tff)/--[bff](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#bff)时会自动启用。

建议：预设值

crop-rect

预设值：无

指定一个位元资料流层级的裁剪矩形。如果想要解码器在播放时裁剪，但因为某些原因不想要裁剪视讯再让x264编码，则可以使用此选项。指定的值是在播放时应该被裁剪的像素。

输入／输出

output

预设值：无

指定输出档名。指定的副档名决定视讯的输出格式。如果副档名无法辨识，则预设输出格式是原始格式（raw）视讯资料流（通常储存为.264副档名）。

特殊位置NUL（Windows）或/dev/null（Unix）指明输出应该被丢弃。这在使用--[pass](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#pass) 1时特别有用，因为唯一在乎的输出是--[stats](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#stats)。

muxer

预设值：auto

指定要输出什么格式。

可用的值：auto, raw, mkv, flv, mp4

auto选项会根据提供的输出档名挑选一个多工器（muxer）。

建议：预设值

参阅：--[output](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#output)

demuxer

预设值：auto

设定x264使用什么解多工器（demuxer）和解码器来剖析输入视讯。

可用的值：auto, raw, y4m, avs, lavf, ffms

如果输入档案有raw、y4m或avs的副档名，则x264会使用相关解多工器来读取档案。标准输入使用原始格式解多工器。否则，x64会尝试以ffms来开启档案，然后再尝试以lavf来开启档案，最后开启失败。

"lavf"和"ffms"选项需要x264以分别的程式库（libraries）编译。如果使用到两者之一，x264会从输入档案带入时间码（timecodes），条件是不能输出为原始格式。这有效地使x264感知VFR。其他选项可以指定--[fps](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#fps)为固定帧率，或者指定--[tcfile-in](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#tcfile-in)为变动帧率。

建议：预设值

参阅：--[input](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#input), --[muxer](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#muxer)

input-csp

预设值：无

告诉x264原始格式视讯输入是什么色彩空间。支援的色彩空间可以从x264 --fullhelp的说明里得知。

注意，虽然有支援RGB色彩空间，但视讯在编码之前会使用bt601（即"SD"）矩阵来转换成YUV。

参阅：--[input-res](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#input-res), --[fps](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#fps)

input-res

预设值：无

指定原始格式视讯输入的解析度。语法是--input-res 720x576。

参阅：--[input-csp](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#input-csp), --[fps](http://www.nmm-hd.org/doc/index.php?title=X264%E8%A8%AD%E5%AE%9A&variant=zh-sg#fps)