DeathTheSheep's Guide to x264 (VFW)

Everything you ever needed to know about x264 encoding. Period.

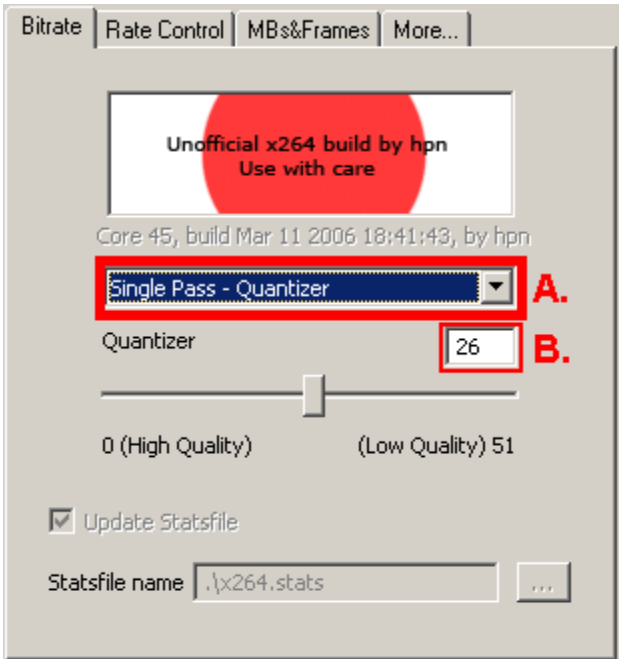
Ever wanted to encode in high-quality AVC video? *DUH, THAT'S WHY YOU CAME HERE.* Basically, you can substitute this guide in just about *any* encoding guide where it talks about configuring XviD. XviD...old, old, XviD...*shudder*... By the way, I'm DeathTheSheep, a poor and unassuming guy and a member of Doom9.org. No more formalities now. **Baaaa!**

Number of Passes

I recommend multipass mode for all of encodes you wish to reach a certain filesize. That's right, if you require your final filesize to be exactly 170.00MB, for instance, you're going to have to do a 2- or 3-pass encode. 1-pass mode is strictly for those who are severely short of time or do not expect to attain high-quality output. 2 passes tends to be sufficient for most people's needs, producing high-quality files with accurate filesizes. However, single-pass mode is sometimes used to produce constant quality output. See the *Constant Quantizer Mode* section below if you wish to encode in a uniform, constant quality regardless of the filesize.

Constant Quantizer Mode

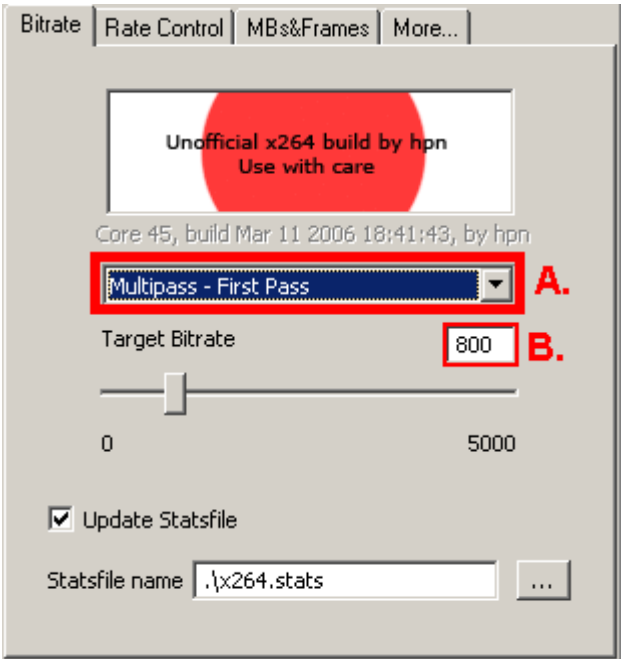
If you don't have to have your videos reach a certain, explicit filesize, or prefer to let the encoder decide what filesize to give it for best quality, you might consider looking into *constant quality* encoding via the “Single Pass - Quantizer” mode.



- Ensure that you have “Single Pass – Quantizer” mode selected from the pass type menu (A.).
- Bear in mind that the *higher* the quantizer (B.), the *lower* the quality of the video will be; thus, a *smaller* filesize.
- Do **not** use a quantizer under **15** unless you are working for archive/reproduction quality.
- Also, do **not** use anything over **40**: although the filesize will be incredibly small, the quality of such an encode is simply unbearable unless you are encoding an extremely sharp video consisting of high-contrast edges.
- A good bet for most people interested in high quality video would be the range of **20** (highest quality) to **30** (lower quality), depending on individual preference and the amount of disk space reserved for the encoded file.
- On animated content such as cartoons and anime with few detailed textures, consider using a higher quantizer.
- On "real-life" content, especially videos with many dark scenes and important subtle textures, consider using a much lower quantizer value.
- Go on to the other tabs to tweak the remainder of the settings.

Multipass Mode

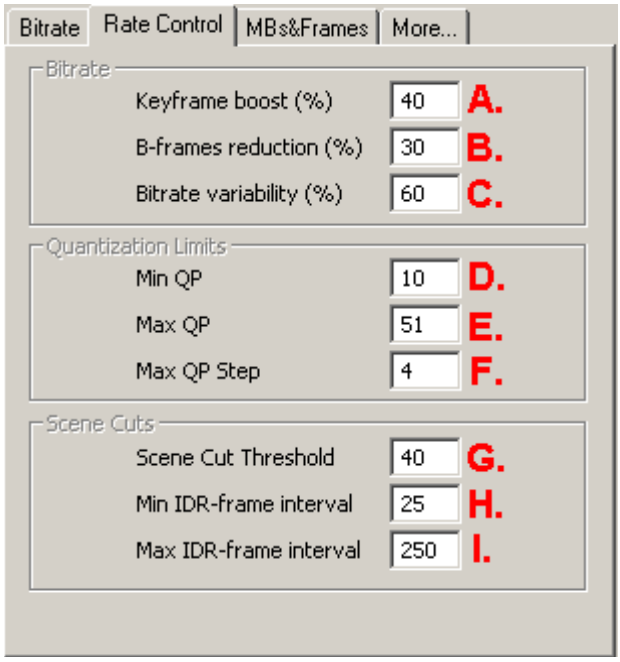
If you wish your video to reach a specific, accurate filesize at the end of the encode, you'll need to use a multipass mode.



- Select “Multipass – First Pass” or “Multipass – First Pass (Fast)” from the pass type menu (A.). For people interested in doing 3 passes for optimal quality, consider the use of the “fast” first pass. For 2-pass encodes, I suggest use of the standard first pass mode. The quality of encodes using fast over standard first pass does not decrease to any catastrophic degree by most standards, but I nevertheless recommend using the standard mode for 2-pass encodes to ensure optimal quality for the largest variety of videos and encoding situations.
- The “Target Bitrate” (B.) in kbps is data rate allocated to the video. The higher the rate, the less the visual information would have to be compressed, leading to better quality, clearer picture, less encoding artifacts, and preservation of more detail. Consider the usage of a bitrate calculator to determine the bitrate in kbps the video must receive to reach the desired filesize (you can find these by hunting for 'em in Doom9's forum, just don't kill any sheep while you're there). I suggest the usage of 500 to 900 kbps for high-quality video, but like I said, it all depends on the filesize you wish to reach. Reasonable filesize limits are 500, 700, or 1400MB for a movie, depending on the length and enthropic complexity of the video (the large amounts of detail and high motion being the chief contributers to high-complexity).
- Go on to the other tabs to tweak the remainder of the settings.
- After tweaking the settings satisfactorily and encoding the first pass, re-enter the codec settings and select “Multipass – Nth Pass” from the pass type menu, keeping all other options the same as they were with the first pass. Repeat this bulleted step once more if you wish to perform a 3-pass encode.

Encoding: High-speed, High-quality

This section provides the steps necessary to produce the **best mix of encoding quality and encoding speed**. Following are the recommended settings which produce the maximum quality AVC encodes while maximizing encoding speed.



- "Keyframe boost (%)" (A.) controls how much the quality of key frames (I-frames, intra frames, the "scene-changer" frames) is boosted above other (P) frames. The boost gives these critical frames some extra quality, so frames predicted from them will also be of better quality, leading to a better visual impression. Additionally, key frames frequently provide the backdrop in many still scenes which may stay on the screen for extended periods of time (especially in slow anime), so it's critically important that these key frames are boosted. Keep this at either **40 or 0** (constant quality, all frames weighted the same) in order to attain what is generally accepted to be the best results. However, tweaking it a bit might help in individual cases (perhaps making it a bit higher in low-bitrate/low-quality anime encodes or lower for "real-life," bright, continuously fast-moving videos where the background is frequently changing or not staying put).
- "B-frames reduction (%)" (B.) controls how much the quantizer (quality) of B-frames is reduced below other (P) frames. This cut in B-frame bitrate is hardly noticeable to the human visual system because B-frames are used inconspicuously in between other (P) frames to more efficiently recreate the video's motion. Setting this lower can substantially increase compression, but you may incur the penalty of motion artifacts and inconsistent detail reproduction. For high-bitrate/low-quantizer "real-life" video, you might want to lower this slightly below the recommended value of **30**, perhaps to **25 or 20**, thereby ensuring a more uniform quality to details, motion and the video as a whole. For animated content (cartoons, anime), you can set this up to **50**, and for some anime material, even all the way up to **60**, due to anime's inherent lack of uniform motion or exquisite detail transformations.
- "Bitrate Variability (%)" (C.), also known as "qcomp," controls the extent to which x264 can fluctuate the quantizers (quality) at the bitrate you specified. The *lower* this option is set, the more unstable and erratic the quantizer (quality) changes become. A value of **0**, for example, would allow x264 to drastically change the quality of each frame: the drawback is that "easy" frames (low-detail, low-motion) frames will look spectacular at your selected bitrate, while other "complex" frames (high-detail, high-motion) look terrible because x264 can instantly lower their quality drastically to keep the desired bitrate, resulting in a very stable, constant bitrate. The *higher* this setting is, however, the more equal the quality becomes, producing a video with more stable quality. Setting this to **100**, for instance, would force x264 to provide the

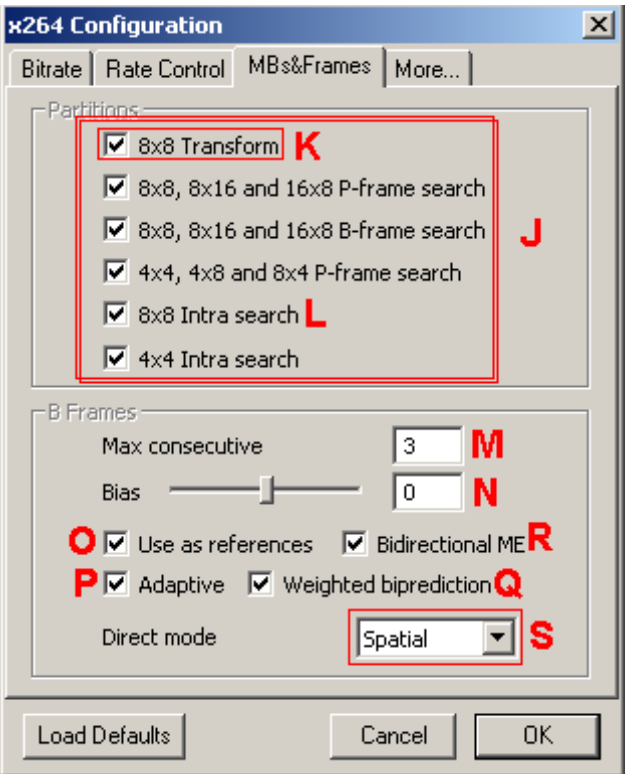
same quality for all frames (no fluctuations), so in theory, it can allot one frame 10 kbps and another 1000 while maintaining equal quality for both (pure VBR); as a whole, therefore, all frames would be quantized uniformly, possessing the same quality due to the restriction in quality fluctuation, even though the bitrate can fluctuate drastically.

  - Think about it this way: this option does exactly what its name says it does. The less you set this "Bitrate Variability," the less your *bitrate* will vary from one scene to the next, but the *quality* will appear more constant.
  - Setting this option is up to personal preference, and I highly suggest you stick with the default of **60** for best quality unless you specifically want a CBR encode (for streaming or something, where you would set this to 0) or true single, equal quality VBR encode (where you would set this to 100).
  - Of course, bitrate variability control is only available in a bitrate controlled mode such as multipass or "Single Pass – Bitrate" mode (which we will not discuss here) and not for Constant Quantizer mode, which is already constant in quality.

- D., E., and F. focus on the so-called *limits of quantization*, also only available for bitrate-controlled modes. For most encodes, you wouldn't have to touch any of these. "Min QP" (D.) sets the lowest quantizer (or the highest quality) your video can attain at any point. I suggest values from 10 to 15 unless you plan on performing an immensely high bitrate encode on a video of low resolution. "Max QP," similarly, sets the highest quantizer (lowest quality) you video can reach. This is even harder to set, and I recommend leaving it where it is (the default of **51**) and let x264 decide what to quantize each scene. A maximum of 51 may seem excessive , but on many pure-black/pure-white frames, it makes perfect sense to the x264. "Max QP Step" controls how quickly x264 can switch between one level of quantization (quality) and another. Setting this too low forces the codec to switch quality levels slowly, which can be devastating to the compression of certain scenes which could suddenly use a lower quantizer to maintain similar visual quality, and setting it too high can lead to conspicuous jumps in quality.
- The "Scene Cut Threshold" (G.) determines how much a frame needs to change before being considered a scene change by x264 (and therefore being awarded a key frame). **40** is the recommended default, but tweaking it per video/encode might help quite a bit in accurately detecting scene changes. For dark/night/cave/underwater/etc videos, you might want to decrease this value so as to let x264 more accurately determine scene changes. As a more general rule, videos with more subtle scene changes need a higher scene cut threshold (such as **45 to 50**), and bright, high-contrast videos with very prominent scene changes might actually be reason to attempt to set a lower threshold (such as **35**).
- The "Min IDR-frame interval" (H.) sets the minimum number of frames that have to go by before x264 can apply a key frame. Usually, the **framerate of the video** you're encoding can be used for this parameter, effectively restricting x264's ability to place key frames by limiting their placement to within 1 second of each other. If this value is set too high, too much time will go by without x264 detecting a scene change and implementing the necessary key frame. Seeking lag can also result. Setting it too low can result in a waste of bitrate and sometimes flickering: too many key frames go by in too short a time period for the eye to notice/appreciate the extra detail allotted to them.
- "Max IDR-frame interval" (I.) sets the maximum number of frames that can go by before x264 has to apply a key frame. Usually, set this value to your **video's framerate x 10**, which would force a keyframe after 10 seconds if none have been placed during that time. Setting this too low can result in an oversaturation of key frames, possible flickering, and general bitrate waste,

reducing overall quality. Setting this too high can result in seeking issues, and any artifacts/encoding flaws during the time between key frames will remain on the screen longer. I believe it's safe, however, to set this up to **1000** or even higher if you don't mind the seeking lag, especially if your scene cut threshold is low enough as it is to stimulate a healthy influx of key frames.

The Next Tab: Mbs&Frames

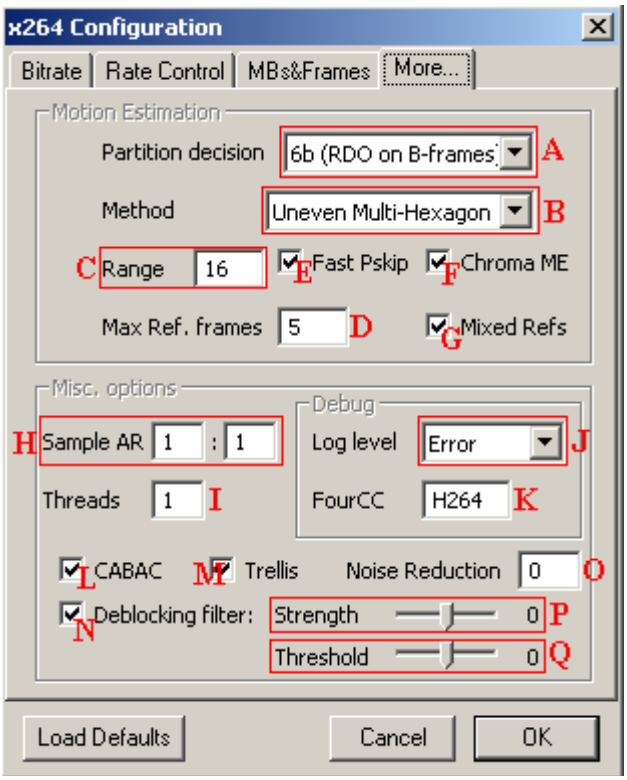


- The partition search and decisions are controlled by the options in box J under the heading "Partitions." These options increase the accuracy and thus the quality and compression efficiency of the codec, enabling higher quality output. As a general rule, the more searches types of searches the codec performs on the visual information, the more accurately and efficiently it can predict and encode them, so I'd recommend you **check all of them**. However, there are is a special case you might need to know about, covered in the next bulleted point.
- The "8x8 Transform" feature (K) is a very powerful compression technique which acts upon larger blocks of visual information and enables the usage of other high-quality partition options such as the "8x8 Intra search" (L) which would otherwise be unusable without the 8x8 transform. Both of these options increase quality to a significant degree, but the usage of the the 8x8 transform would make your video "High Profile AVC" and would break the encoded video's compatibility with "Main Profile AVC." Without this compatibility, your video might not play back with certain decoders which don't support "High profile AVC." Currently, most of the popular or highly developed desktop decoders all support high profile, so it's safe to use this feature for playback on your desktop. See the *Compatibility* section below for information on device compatibility.
- The maximum amount of B-frames x264 can use in a row is controlled by option M. B-frames

are highly compact frames which are placed in between P-frames (standard frames predicted from the I or key frames) to serve as an efficient predicted model of this frame in between. These increase compression efficiency to a great degree with very little visual impact, even though they are typically assigned a higher quantizer than other frames (See option B above for more info). As long as option P ("Adaptive" check box) is checked, I suggest you raise this limit to **3** or higher, so that x264 can decide for itself the best number of B-frames to use to keep maximum quality.

- B-frame "Bias" (N) is used to tweak x264's usage of B-frames. The higher this number, the more frequently x264 will decide to drop in B-frames. Setting this to 100, for instance, will force x264 to use the maximum number of B-frames specified in box M; however, this result can be attained simply by unchecking "Adaptive" (P). Setting this to -100 means the use of hardly any B-frames at all—none, in fact, for most situations. For most videos, the optimal number of B-frames x264 choses to use at bias **0** is typically the best choice, and I'd recommend keeping it here.
- The x264 codec is capable of using the B-frame pyramid feature to "Use [B-frames] as references" (O). To use B-frames as references for other consecutive B-frames increases quality slightly because these B-frames can be predicted from one another instead of being limited to the P frames around them for prediction. B-pyramid is most efficient with the usage of 3 max consecutive B-frames (M). I recommend that this option be turned on, as it slightly increases quality at little encoding speed loss.
- "Weighted biprediction" (Q), when checked, allows B-frames to be predicted more heavily from one P frame or another. It results in more accurate and efficient B-frames, therefore increasing quality. I highly recommend you check this.
- "Bidirectional ME" (R) is a newly feature which allows x264 to predict some B-frames using motion before and after them. This increases quality. Use it.
- Direct B-frame mode (S) allows B-frames to use "predicted motion vectors" to be used instead of coding the actual motion, thus saving space and increasing compression efficiency. Of the 2 modes currently available, I suggest usage of Spatial for animated content (better handling of inconsistent motion jumps) and Temporal for real-life content (fluid motion).

The Next Tab: More...



- The method of "Partition Decision" (A), also known as "Subpixel Refinement Quality," is a crucial feature which controls the extent to which x264 will go to come up with the best "decisions" regarding motion estimation. The selection menu contains a list of 7 options, arranged from the least quality "1 (Fastest)" to the highest quality "6b (RDO on B-frames)" options, in order. Bear in mind that using higher quality options results in x264 "pondering" more about which decisions to make, resulting in a greater speed penalty. Since this is one of the biggest "make it or break it" quality options in x264, make sure to set it properly. I suggest using values **no lower than '5'**. Ever. Period. '5' is very fast and allows "Chroma ME" to be used (see F below). "6 (RDO)," however intimidating it may seem, is extremely beneficial to encoding quality because it enables *rate distortion (RDO)*, a feature that drastically increases decision quality (and therefore compression quality) at the expense of taking more encoding time. '6b (RDO on B-frames)' enables this rate distortion decision on B-frames too, giving them a jump in quality and further lowering encoding speed. In my opinion, however, the drastic quality gains attained by use of rate distortion ('6' and '6b') greatly outweigh the speed lost during the encoding process, and for that reason, especially if you have a fast computer, I recommend the use of **"6b (RDO on B-frames)"** partition decision.
- The motion estimation "Method" option (B) allows the user to select how x264 looks for motion. The better the method, the higher the likelihood of x264 finding and accurately recording motion, thereby increasing the compression quality and efficiency. I suggest use of the **"Hexagonal search"** (searches horizontally, vertically, and diagonally) for slower machines or people short on time, but otherwise the **"Uneven Multi-Hexagon"** (searches via multiple hexagonal bursts) search is superior, providing better quality due to its more accurate search algorithm, despite an increase in encoding time. *NEVER* even think about using "Exhaustive Search" (pixel-by-pixel search, no "estimation" at all); it's not much better than "Uneven Multi-Hexagon" and it decreases encoding speed so drastically you'd be lucky if your encode finished

in a week. I don't even know why it's offered as an option.

- Notice that the user can specify the search's "Range" in pixels (C) with the usage of "Uneven Multi-Hexagon" or (please don't) "Exhaustive Search." Keep this at **16**: that's what the algorithm was optimized for: it usually provides the best speed and quality. However, if you have high-resolution content with a low frame rate (under 15), consider increasing this a bit, perhaps to 24 or an absolute maximum of 32. Also, if you have low-resolution content with exorbitantly high frame rate (over 60), you might think about cutting this down to 10.
- You can set the number of references x264 can use via the "Max Ref. Frames" option (D). AVC can efficiency encode new frames by making reference to similar frames it previously encoded. The maximum number of frames that can be referred to ("referenced") is set by this option. The higher the better, the higher the slower. For real-life content, **3 to 5** is sufficient in most cases while maintaining high encoding speed. For animated content, or content with redundant or cyclical motion, more references can help significantly, so here I'd suggest using up to **8 or 10**, providing "Mixed Refs" (G), which allows x264 greater freedom to make references on a smaller scale, is checked.
- "Fast Pskip" (E) is a means by which x264 speeds up the encoding process, but it occasionally leads to artifacts in flat scenes or subtle gradients. If this is a problem for you (and you notice artifacts in these places), unchecking "Fast Pskip" might help. There is very minuscule overall quality gain attained by disabling this (forces codec to examine each block thoroughly), and a small but significant speed loss. Therefore, I recommend leaving this option **unchecked unless necessary.**
- "Chroma ME" (F) works by optimizing color during motion estimation. This almost always leads to significant quality increase, especially with animated material. I recommend enabling (checking) this option.
- The "Sample AR" (H) deals with the aspect ratio of the video-- if you don't know what it does, you don't need to touch it ;)
- Set "Threads" (I) to the number of encoding threads x264 should allocate to the encoding process. The number of threads you should use corresponds with the number of processor cores your computer has. Remember, count each single HT processor as 2 threads. If you don't know how many cores your computer has, or whether or not it has HT (hyper threading), it's safe to keep this at **1**. The only benefit of selecting more cores is that it speeds up the encoding process on multi-core machines.
- Debug logging (J): records information about the encoding process. Keep this on **Error**—it serves no purpose to the average user to amass any further debugging information.
- The FourCC (K) of the AVI file x264 produces is the identifier code of the type of video stream inside it. XviD produces AVI files with the "XviD" FourCC unless specified otherwise. H264 is a widely accepted and supported FourCC for AVC files, as is AVC1. I suggest you leave it at the default of **H264**.
- CABAC a.k.a Context Adaptive Binary Arithmetic Coding (L) is a key feature of x264 main profile allowing syntax elements of the video stream to be predicted by context. Basically, it increases compression at no quality loss (10-15% varying by video and by bitrate), though your *decoding* speed will suffer slightly (usually high bitrates [and/or low quantizers because of bitrate impacts] cause complex CABAC decoding). If disabled, x264 resorts to the less-quality CAVLC, and you will lose access to some other encoding options (trellis, notably). I highly recommend you keep this option **checked** unless you want the fastest possible decoding speed

(XboX, handheld, etc).

- "Trellis" (M) is a high-level feature which increases compression quality (this is subjective and varies by person/bitrate/source-type) by manipulating what data is kept in the encoded file late in the compression process. It supposedly increases the efficiency of the bitrate used, but due to its massive encoding speed (~10%), it's unpredictability in Constant Quantizer mode, and the relatively small (even negative) quality gains often reported, I suggest leaving this off during Constant Quantizer mode and only on in multipass encodes if you have a fast computer or want to milk the codec for all it's worth.
- Consult the *Deblocking Guide* below for information on how to correctly configure the inloop "Deblocking Filter" (N).

Deblocking Guide

Inherent in the AVC format is an exceptionally useful feature which reduces blocking and other encoding artifacts that so plagued XviD/DivX and the other ASP predecessors of AVC. It is extremely useful in AVC encoding, and due to the fact that AVC is built around this filter and you may experience excessive amounts of blocking and video artifacts (bad stuff), it should never be disabled for normal use. However, due to the fact that it takes up immense amounts of decoding time, you may consider disabling it to achieve the fastest possible decoding speed (Xbox, handheld, etc) if the decoder doesn't support disabling it on the fly.

Bear in mind that the very principle of in-loop filtering (deblocking) is rather controversial due to the fact that the human visual system (HVS) often misinterprets the flaws and blocks in spatially complex scenes as the scene's actual *detail*. Because x264 removes this artifacting by default, the HVS perceives this to be a "loss of detail" even though *technically* x264's detail is just as accurate—even more so, in most cases, at similar bitrates. Indeed, the general consensus as of yet (heavily supported by Doom9's codec comparison and the quality metrics) is that x264 does tend to keep more detail more accurately than ASP (non-deblocked) codecs at similar bitrates. However, the deblocker's removal of artifacts is occasionally confused by the HVS as a removal of legitimate *detail* by people *accustomed to encoding without a deblocking filter*.

The deblocking threshold (See 'Q' in above illustration) determines how much of the material actually needs to be deblocked. What is a block? That's what the threshold attempts to figure out. The higher this is set, x264 will perceive more of the video as blocks (often too much of the video, if set too highly), so the more the deblocker will act upon. The deblocking strength (P) determines how strong of a deblocking effect is needed to eliminate the blocks where the threshold identified them. Naturally, if you don't wash (deblock) enough of the material, there will still be some stains (blocks) left, away from the washed area. Think of it like washing a shirt. If you don't wash (deblock) hard or strong enough, the stains (blocks) simply won't fade/disappear. However, if you wash too much material or scrub too hard, you'll ruin the shirt, because instead of stains, there will be the ugly lack of color where all the texture and detail rubbed out.

- Both bars are initially set to **0** for a reason: this is the standard deblocking that will lead to the highest quality balance block removal while maintaining detail. If, however, you find the result unsatisfying, look to these tips:
- To determine what you think is best, try the standard **0/0** settings on a small but indicative sample of the source and experiment with it, keeping in mind the below guidelines.
- For deblocking strength and threshold both, try not go out of bounds of the **-3 to 2** range.

Generally, any more than 2 will turn your result into mush while decreasing quality. Any less than -3 may cause the result to look a bit too sharp—not in a good sense, either, for any lack of texture will merely become more apparent as all smoothing is taken out.

- If you are encoding an animated source, heavier deblocking (both bars at **1**) is suggested to eliminate all blocks possible. The drawn content is more resistant to smearing due to the high-contrast edges. On the other hand, if you are encoding a "real-life" video, especially one with intricate textures and low/poor lighting, consider decreasing the deblocking to preserve such things without creating a washing effect.
- Try to keep a positive correlation between the two settings. That is, if you want heavier deblocking, make sure to increase the threshold so that more gets deblocked, and vise versa. Recall the comparison with the clothes washing: you don't want to heavily wash a small area while the rest remains unwashed; the unwashed areas will stand out more in stark vividness and provide an ugly visual effect.
- For maximum preservation of detail at high bitrates, I suggest trying **-1** for both settings. However, **0** may still be optimal for one or both of the settings depending on source material (gradients, textures and contrasts), bitrate used, and personal preference.