

# Quake 3 Engine – Smoothness Guide

([http://www.mikemartin.co/quake3/smoothness\\_guide](http://www.mikemartin.co/quake3/smoothness_guide))

([http://www.mikemartin.co/gaming\\_guides/quake3\\_smoothness\\_guide](http://www.mikemartin.co/gaming_guides/quake3_smoothness_guide))

This guide will explain how you can make your quake 3 running the way it was made/should run.

This is also assuming you do not have a wireless connection or a 56k connection or cant even hold 125fps solid. otherwise don't bother with half of these settings.

These settings are also with complete disregard to personal preference and could possible hinder your abilities. so experiment with what works for you and for different pings.

## RATE 25000 (25kbps)

Watch your lagometers green spikes when this is set lower. mostly everyone can handle this it just depends on how solid your connection is, routing, the server, etc..

## SNAPS 40

40 is a general number because the server will lower it to the servers sv\_fps for you. cpma forces your snaps to equal the servers sv\_fps when you connect so dont bother setting a value

## R\_SWAPINTERVAL

0: Off

1: TRUE vsync in quake 3, it LOCKS your fps with your monitors refresh rate. So, those of you that know what your doing, OVERRIDE your monitors refreshrate to 125hz at whatever resolution you play in, most of you should be able to get 120hz with a crt. And if you can't get over 100hz with your monitor, you'll probably experience mouse lag.

Test by slowly upping the refresh rate and going ingame, if your screen shuts off, hit the console key and type /quit and hit enter. You'll have to do this blind so be carefull :)  
example: say your desktop resolution is 1024x768 but you play in 800x600, in your video cards profile override the refreshrate for 800x600 so when quake 3 loads, your screen resolution bumps down from 1024 to 800 and that raised refreshrate gets used automatically.

- if you dont notice a difference, then you'll love q4

## CL\_MAXPACKETS \*\*\* -With R\_swapinterval 1

\*\*\* equals your FRAMES PER SECOND(monitors refresh rate), if you set this to ANY NUMBER lower than this, it will CUT YOUR MAXPACKETS IN HALF.

example: FPS =125 CL\_MAXPACKETS =100 means your MAXPACKETS BECOME 63 (half your fps because 100 is NOT divisible by 125)

## CL\_MAXPACKETS \*\*\* -With R\_swapinterval 0

\*\*\* equals your COM\_MAXFPS, if you set this to ANY NUMBER lower than your com\_maxfps, it will CUT YOUR MAXPACKETS IN HALF.

example: COM\_MAXFPS =125 CL\_MAXPACKETS =100 means your MAXPACKETS BECOME 63 (half your fps because 100 is NOT divisible by 125)

## IN\_MOUSE

### IN\_MOUSE -1

Captures the position the Windows cursor and passes that info to the game to change the camera angle. Once that is done for a frame, the cursor is returned to the center of the screen. There is nothing else applied to the data until it gets to the sensitivity, acceleration, m\_yaw, and m\_pitch part of the code.

#### Pros:

As responsive as you can get

It's accurate

Perfect for Logitech's SetPoint Implementation Option

#### Cons:

The cursor can clip the edge of the window and not be able to go further, resulting in negative acceleration. The higher your Q3 resolution and framerate, the less you will run into this problem (800x600 is pretty much safe for any sens, 640x480 might have some clipping with low sens). Higher mouse sampling rates or r\_finish should also be used to prevent Q3 from getting 2 or more large mouse samples in one frame.

Must use CPL mouse fix (or alternative) to eliminate Windows enabling Pointer Precision automatically for games in WinXP.

Some may start to miss jumps set to -1

### IN\_MOUSE 1

Uses DirectInput to directly access the mouse, bypassing the windows cursor. It also applied ~12ms worth of mouse smoothing automatically before reaching the sensitivity and acceleration part of the Q3 code.

#### Pros:

No negative acceleration

Works independantly of Windows cursor settings (i.e. Sensitivity and Pointer Precision)

#### Cons:

Uncontrollable mouse smoothing which adds latency to mouse actions.

Delay when clicking mouse (click -> wait 12ms -> shoot = bad)

Even raising your mouserate (hz) even makes a large difference, not so much in new mice since alot come set to around 500hz. Use cl\_showmouserate 1 to see ingame.

## R\_DISPLAYREFRESH Xxx

Set to your monitors refreshrate. ie: 120hz,75hz, etc..

## CG\_PREDICT

- 0 - off
- 1 - normal / default
- 2 - optimised

The normal prediction path is extremely slow at times (notably around curves) and can cost you 100fps on a GHz machine. This new scheme is MUCH faster, but slightly more prone to errors. Oddly enough, it's still more accurate than the original id prediction code (i.e. before the CPMA fixes). If you have a slow machine, it's definitely worth trying. Note that cg\_predict 2 was introduced 9 Sep 2002, the definition of a "slow machine" has changed since then. Most of today's computers will not notice any difference at all.

Note: Do not use cg\_predictItems 1 with cg\_predict 2.

## CG\_SHOWPLAYERLEAN

0: Allows the disabling of viewed model leaning that was introduced in the 1.27x patches. Players appear to "lean" in the direction they are walking in so its easy to see them change directions.

1: Could be used to help judged where the persons moving next. Then again it can throw you off.

## CL\_PACKETDUP 0-3

0 uses the least ammount of bandwidth and doesn't "waste" time sending packets twice. Set higher if you have much packetloss. You can check your packetloss easily by using cg\_predict 1. I use cl\_packetup 2 or 3 is about as high as i would go.

## CG\_OPTIMISEBW

MUST be 1. Don't ever use 0, and only use 2 if your connection is REALLY bad, ie 56k or below.

## CG\_NUDGE

Keep at 0 if you ping below 90-100. If you ping higher use negative nudge(not recommending due to extremely choppy players, almost the same as cl\_timenudge)

## CL\_TIMENUDGE

This still exists in CPMA, but should always be 0 unless you're so used to "normal" Q3 netcode that you've become dependent on it. All it really does now is mess up the automatic adaptive nudges.

## CG\_XERPCLIENTS

-1 - Hacked extrapolation: intended for HPBs. This smooths players out when you use high timenudges, at the cost of some accuracy. It's typically easier to hit a smooth target that's a few pixels misplaced than it is to hit one that looks like it's teleporting all over the map, so this combined with cg\_nudge is the best option for HPBs.

0 - No extrapolation. Fine you're LPB.

1 - Can be used but errors will occur (players in walls and stuff)

## CG\_LAGHAX

-1 is nice, shouldn't be changed unless you have ping over 100 and use negative nudge.

## DISABLE PUNKBUSTER -CPMA ONLY

Most cpma servers do not run punkbuster, you need to know cuz you need to know.

```
seta cl_punkbuster 0  
pb_cl_disable
```

---

## The Confusing Section

### RATE

Measured in Bytes per second. This is the maximum rate at which you will be able to receive data from the server. You should set this to the highest downstream rate your own connection can achieve consistently. This should not be set higher than 25000. For most of you this will be set at 25000, but for others with slower connections, you can take the rate at which you connect at and divide it by 10 to get your Rate value. If you set your rate higher than what your connection can actually achieve, this is what will happen:

When there are more players on the server (or just players constantly shooting) there is more data that needs to be sent from the server to the clients. If you have your rate set too high, the server will try to take advantage of your extra bandwidth. So when the server tries to send you 250Kbps down your 36Kbps connection, you will be flooded, your ping will 999, you will get mass packetloss and your connection will either hang or you will get kicked by the server.

### SNAPS

This setting is the number of 'snapshots' that you receive from the server per second. A snapshot is a 'picture' of what is happening in the game since the last snapshot. This is the data transmission equivalent of a framerate.

This setting should be set equal to the server's sv\_fps setting (default is 20). Some servers use sv\_fps 30 and I have seen some sv\_fps 40 servers. I have never seen a server using a sv\_fps greater than 40. Regardless of how high you set your snaps setting, you will not be able to receive more than the rate at which the server runs which is defined by its sv\_fps setting. So you can set your snaps to 40 and play on a sv\_fps 30 server without needing to adjust your snaps - it will run at 30 anyway.

If you have dialup or wireless you will probably be better using snaps 20. The higher the snaps the more bandwidth will be needed, so snaps 30 or 40 may cause the connection to be less stable.

## CL\_TIMENUDGE

This affects how your client processes the snapshots it receives. Normally the client compares one snapshot with the next and interpolates between the two. If you set this to a negative value, you can adjust the interpolation of snapshots to compensate for lag by making your client start interpolating BEFORE it receives the next snapshot. In other words, it will be predicting what the next snapshot will look like and using that prediction as a basis for the interpolation. At some point during this nudged interpolation, the actual next snapshot will arrive from the server. At this point, the client will replace the predicted snapshot by the actual snapshot and recalculate the interpolation. This difference between the two is the prediction error. However at this point, you will not see a huge jerk due to the change from predicted to actual data as the prediction error is decayed away to ensure the gameplay remains as smooth as possible, even if it is slightly inaccurate.

The more negative timenudge you use, the more predictions will happen and therefore the more inaccurate the predicted data will be. Therefore, even though these errors are decayed, the jerks will be more noticeable and as a result other players will seem to move less smoothly.

Effectively, the greatest negative value is equal to  $1000/\text{sv\_fps}$  (of the server) however -12 is a recommended maximum to keep the gameplay smooth and accurate, even if there is a little less lag compensation.

There is also another side to the timenudge story. You can use positive values of timenudge to affect snapshot interpolation with an opposite effect. Using positive values means that your client will wait for extra time before interpolating the 2 snapshots. This can be useful if you have a higher ping or a packetlossy connection. In these cases snapshots are likely to be lost and the time between each snapshot arriving can fluctuate greatly as some packets are delayed or whatever. In this case, even if you are not using any timenudge, the client may have to start interpolating before the next snapshot has arrived, simply because it is late. By using a positive timenudge, you can ensure that the client waits those crucial few milliseconds more to allow the next snapshot to arrive successfully, thus keeping gameplay smooth and accurate - even if the incoming data is deliberately lagged.

Btw using timenudge will not affect your ping on the screen, simply because your ping has not changed. Timenudge has nothing to do with your actual ping, but it has everything to do with your lag. I would like to add that timenudge only directly affects the lag of the incoming data and does not directly affect the data that you send to the server. Whether timenudge indirectly affects your outgoing data is unlikely but that issue is best being left for a timenudge-specific topic.

## CL\_MAXPACKETS

This is the Maximum number of data packets that can be SENT to the server per second. The higher you set this, the smoother your game will feel as your actions (run, shoot, jump etc.) will be updated more frequently. Broadbanders will be able to go up to 125 (but not higher) but this depends on your individual connections. Dialup users will just about be able to manage 30, so 20 is more of a stable setting.

Also if you set your MaxPackets too high, your ping will start to rise. This is due to the increase in upstream bandwidth usage. So you should set your MaxPackets as high as you can without it affecting your ping (and without going over 125).

COM\_MAXFPS

This is the maximum graphical framerate permitted. You can not use any MaxFPS value. The only valid values are those which are equal to  $(1000/x)$  where  $x$  is an integer. So for example your 125fps comes from  $(1000/8 = 125)$ . If you try and set MaxFPS to 120, you will still get 125fps. This is because any invalid setting is rounded up to the next  $(1000/x)$ .

Some valid Values for MaxFPS:

$1000/3 = \sim 333$

$1000/4 = 250$

1000/5 = 200  
1000/6 = ~166  
1000/7 = ~142  
1000/8 = 125  
1000/9 = ~111  
1000/10 = 100  
1000/11 = ~90  
1000/12 = ~83  
1000/13 = ~76  
1000/14 = ~71  
1000/15 = ~66  
1000/16 = ~62  
1000/17 = ~58  
1000/18 = ~55  
etc.

## CL\_MAXPACKETS AND COM\_MAXFPS

This is where your graphical framerate affects your connection. The actual MaxPackets sent to the server MUST be equal to (your\_FPS/x), once again where x is an integer. If you set your MaxPackets to a value which is not equal to (your\_FPS/x) then your MaxPackets will actually be rounded down to the nearest valid setting.

Most of you will use MaxFPS 125. If so, your valid MaxPackets values will be as follows (assuming your PC can achieve its MaxFPS value constantly):  
(always round up these calculations to integers)

125/1 = 125  
125/2 = 63  
125/3 = 42  
125/4 = 32  
125/5 = 25  
etc.

You should choose the largest value that your connection can handle. If your MaxPackets is not set to a valid value, it is a potential waste of bandwidth and therefore potential lag/instability. For example, if your framerate drops to 100 then your valid MaxPackets values will change to:

100/1 = 100  
100/2 = 50  
100/3 = 34  
100/4 = 25  
etc.

So if you have your Cl\_MaxPackets set to 100 and are using 125fps then mostly your MaxPackets will be at 63 (this is the largest value not exceeding the input Cl\_MaxPackets limit). However, when the framerate drops to 100fps, the MaxPackets will change to 100 as this is now the largest value that does not exceed the input Cl\_MaxPackets limit. If your connection cannot handle 100 packets it will cause your ping to rise or spike. Even if it can handle the jump from 63 to 100 packets, this may well cause your ping to fluctuate more than if your actual MaxPackets were constant.

\*Oh and having maxfps 333 can mess up a server by causing everyone else to lag. I don't know why or under what circumstances, but I've seen it happen when two or three players with 333fps get on the same server.