

# ETQW CVAR List

| <b>cvar</b>               | <b>Description</b>   |
|---------------------------|--|
| a_glowBaseScale           | Original image contribution factor<br>type: float (cheat)  |
| a_glowLuminanceDependency | Dependency of the glow on the luminance(brightness)<br>type: float (cheat)   |
| a_glowScale               | Blurred image contribution factor<br>type: float (cheat)   |
| a_glowSpriteMin           | type: string (cheat)   |
| a_glowSpriteSize          | type: float (cheat)  |
| a_glowSunBaseScale        | Factor to scale to sun factor with<br>type: float (cheat)  |
| a_glowSunPower            | Power to raise to sun factor to<br>type: float (cheat)   |
| a_glowSunScale            | Factor to scale to sun factor with<br>type: float (cheat)  |
| a_glowThresh              | Threshold above which part of the scene starts glowing<br>type: float (cheat)  |
| a_sun                     | type: float (cheat)  |
| a_windTimeScale           | Speed at which wind effects change<br>type: float (cheat)  |
| aas_extendFlyPaths        | set to 1 to enable extending fly paths<br>type: bool (cheat)<br>default: 1 (true)                                    |
| aas_locationMemory        | used to remember a particular location, set to 'current' to store the current x,y,z location<br>type: string (cheat) |
| aas_optimizePaths         | set to 1 to enable path optimization<br>type: bool (cheat)<br>default: 1 (true)                                      |
| aas_pullPlayer            | pull the player to the specified area<br>type: string (cheat)  |
| aas_pullWalk              | walk instead of sprint<br>type: bool (cheat)<br>default: 0 (false)   |
| aas_randomPullPlayer      | pull the player to a random area<br>type: bool (cheat)<br>default: 0 (false)   |
| aas_showAreas             | show the areas in the selected aas<br>type: bool (cheat)<br>default: 0 (false)                                       |
| aas_showEdgeNums          | show edge nums<br>type: bool (cheat)<br>default: 0 (false)   |
| aas_showFloorTrace        | show floor trace<br>type: bool (cheat)<br>default: 0 (false)   |
| aas_showHopPath           | show hop path to specified area<br>type: string (cheat)  |

| <b>cvar</b>               | <b>Description</b>  |
|---------------------------|---|
| aas_showNearestCoverArea  | show the nearest area with cover from the selected area (aas_showHideArea 4 will show the nearest area in cover from area 4)<br>type: int (cheat) |
| aas_showNearestInsideArea | show the nearest area that is inside<br>type: bool (cheat)<br>default: 0 (false)  |
| aas_showObstacleAvoidance | shows obstacles along paths<br>type: bool (cheat)<br>default: 0 (false)   |
| aas_showPath              | show the path to the walk specified area<br>type: string (cheat)  |
| aas_showPushIntoArea      | show an arrow going to the closest area<br>type: bool (cheat)<br>default: 0 (false)   |
| aas_showTravelTime        | print the travel time to the specified goal area (only when aas_showAreas is set)<br>type: int (cheat)  |
| aas_showWallEdgeNums      | show the number of the edges of walls<br>type: bool (cheat)<br>default: 0 (false)   |
| aas_showWallEdges         | show the edges of walls, 2 = project all to same height, 3 = project onscreen<br>type: int [0, 3] (cheat)   |
| aas_skipObstacleAvoidance | ignore all dynamic obstacles along paths<br>type: bool (cheat)<br>default: 0 (false)  |
| aas_subSampleWalkPaths    | set to 1 to enable walk path sub-sampling<br>type: bool (cheat)<br>default: 1 (true)  |
| aas_test                  | select which AAS to test<br>type: string [aas_player, aas_vehicle] (cheat)  |
| af_contactFrictionScale   | scales the contact friction<br>type: float (cheat)  |
| af_forceFriction          | force the given friction value<br>type: float (cheat)   |
| af_highlightBody          | name of the body to highlight<br>type: string (cheat)   |
| af_highlightConstraint    | name of the constraint to highlight<br>type: string (cheat)   |
| af_jointFrictionScale     | scales the joint friction<br>type: float (cheat)  |
| af_maxAngularVelocity     | maximum angular velocity<br>type: float (cheat)   |
| af_maxLinearVelocity      | maximum linear velocity<br>type: float (cheat)  |
| af_showActive             | show tree-like structures of articulated figures not at rest<br>type: bool (cheat)<br>default: 0 (false)  |

| <b>cvar</b>              | <b>Description</b>  |
|--------------------------|---|
| af_showBodies            | show bodies<br>type: bool (cheat)<br>default: 0 (false)   |
| af_showBodyNames         | show body names<br>type: bool (cheat)<br>default: 0 (false)   |
| af_showConstrainedBodies | show the two bodies constrained by the highlighted constraint<br>type: bool (cheat)<br>default: 0 (false) |
| af_showConstraintNames   | show constraint names<br>type: bool (cheat)<br>default: 0 (false)   |
| af_showConstraints       | show constraints<br>type: bool (cheat)<br>default: 0 (false)  |
| af_showInertia           | show the inertia tensor of each body<br>type: bool (cheat)<br>default: 0 (false)                          |
| af_showLimits            | show joint limits<br>type: bool (cheat)<br>default: 0 (false)   |
| af_showMass              | show the mass of each body<br>type: bool (cheat)<br>default: 0 (false)                                    |
| af_showPrimaryOnly       | show primary constraints only<br>type: bool (cheat)<br>default: 0 (false)                                 |
| af_showTimings           | show articulated figure cpu usage<br>type: bool (cheat)<br>default: 0 (false)                             |
| af_showTotalMass         | show the total mass of each articulated figure<br>type: bool (cheat)<br>default: 0 (false)                |
| af_showTrees             | show tree-like structures<br>type: bool (cheat)<br>default: 0 (false)                                     |
| af_showVelocity          | show the velocity of each body<br>type: bool (cheat)<br>default: 0 (false)                                |
| af_skipFriction          | skip friction<br>type: bool (cheat)<br>default: 0 (false)   |
| af_skipLimits            | skip joint limits<br>type: bool (cheat)<br>default: 0 (false)   |
| af_skipSelfCollision     | skip self collision detection<br>type: bool (cheat)<br>default: 0 (false)                                 |
| af_testSolid             | test for bodies initially stuck in solid<br>type: bool (cheat)<br>default: 1 (true)                       |

| <b>cvar</b>                | <b>Description</b>  |
|----------------------------|---|
| af_timeScale               | scales the time<br>type: float (cheat)  |
| af_useImpulseFriction      | use impulse based contact friction<br>type: bool (cheat)<br>default: 0 (false)                |
| af_useJointImpulseFriction | use impulse based joint friction<br>type: bool (cheat)<br>default: 0 (false)                  |
| af_useLinearTime           | use linear time algorithm for tree-like structures<br>type: bool (cheat)<br>default: 1 (true) |
| af_useSymmetry             | use constraint matrix symmetry<br>type: bool (cheat)<br>default: 1 (true)                     |
| ai_debugAnimState          | displays animState changes for the specified monster entity number<br>type: int (cheat)       |
| ai_debugMove               | draws movement information for monsters<br>type: bool (cheat)<br>default: 0 (false)           |
| ai_debugScript             | displays script calls for the specified monster entity number<br>type: int (cheat)            |
| ai_debugTrajectory         | draws trajectory tests for monsters<br>type: bool (cheat)<br>default: 0 (false)               |
| ai_fallTime                | Number of seconds before the player plays the falling animation<br>type: float (cheat)        |
| anim_forceUpdate           | type: bool (cheat)<br>default: 0 (false)  |
| anim_maxBodyPitch          | max pitch of body adjustment<br>type: float (cheat)   |
| anim_minBodyPitch          | min pitch of body adjustment<br>type: float (cheat)   |
| anim_reduced               | type: bool (cheat)<br>default: 1 (true)   |
| anim_showMissingAnims      | Show warnings for missing animations<br>type: bool (cheat)<br>default: 0 (false)              |
| aor_animationCutoffScale   | scale the aor animation cutoff distance<br>type: float  |
| aor_ikCutoffScale          | scale the aor ik cutoff distance<br>type: float   |
| aor_physicsCutoffScale     | scale the aor physics cutoff distance<br>type: float  |
| aor_physicsLod1StartScale  | scale the aor physics lod 1 distance<br>type: float   |
| aor_physicsLod2StartScale  | scale the aor physics lod 2 distance<br>type: float   |

| <b>cvar</b>               | <b>Description</b>  |
|---------------------------|---|
| aor_physicsLod3StartScale | scale the aor physics lod 3 distance<br>type: float   |
| bot_aimSkill              | Sets the bot's default aiming skill. 0 = EASY, 1 = MEDIUM, 2 = EXPERT, 3 = MASTER<br>type: int  |
| bot_breakPoint            | Cause a program break to occur inside the bot's AI<br>type: bool (cheat)<br>default: 0 (false)  |
| bot_canDropDeployables    | 0 = bots dont drop deployables of any kind, 1 = bots can drop all deployables<br>type: bool (cheat)<br>default: 1 (true)  |
| bot_canStrafeJump         | 0 = bots can't strafe jump, 1 = bots CAN strafe jump to goal locations that are far away<br>type: bool (cheat)<br>default: 0 (false)  |
| bot_debug                 | Debug various bot subsystems. Many bot debugging features are disabled if this is not set to 1<br>type: bool (cheat)<br>default: 0 (false)                                    |
| bot_debugAirVehicles      | Debug bot air vehicle usage. -1 = disable<br>type: int (cheat)  |
| bot_debugGroundVehicles   | Debug bot ground vehicle usage. -1 = disable<br>type: int (cheat)   |
| bot_debugMapScript        | Allows you to debug the bot script.<br>type: bool (cheat)<br>default: 0 (false)   |
| bot_debugObstacles        | Debug bot obstacles in the world<br>type: bool (cheat)<br>default: 0 (false)  |
| bot_debugPersonalVehicles | Only used for debugging the use of the husky/icarus.<br>type: bool (cheat)<br>default: 0 (false)  |
| bot_debugSpeed            | Debug bot's move speed. -1 = disable<br>type: int (cheat)   |
| bot_debugWeapons          | Only used for debugging bots weapons.<br>type: bool (cheat)<br>default: 0 (false)   |
| bot_doObjectives          | 0 = bots let the player play the hero, with the bots filling a supporting role, 1 = bots do all the major objectives along with the player<br>type: bool<br>default: 1 (true) |
| bot_drawActionDist        | How far away to draw the bot action info. Default is 2048<br>type: float (cheat)  |
| bot_drawActionGroupNum    | Filter what action groups to draw with the bot_drawAction cmd.<br>-1 = disabled.<br>type: int (cheat)   |
| bot_drawActionNumber      | Draw a specific bot action only. -1 = disable<br>type: int (cheat)  |

| <b>cvar</b>               | <b>Description</b>  |
|---------------------------|---|
| bot_drawActions           | Draw the bot's actions.<br>type: bool (cheat)<br>default: 0 (false)   |
| bot_drawActionSize        | How big to draw the bot action info. Default is 0.2<br>type: float (cheat)  |
| bot_drawActiveActionsOnly | Draw only active bot actions. 1 = all active actions. 2 = only GDF active actions. 3 = only Strogg active actions. Combo actions, that have both GDF and strogg goals, will still show up.<br>type: int (cheat) |
| bot_drawClientNumbers     | Draw every clients number above their head<br>type: bool (cheat)<br>default: 0 (false)  |
| bot_drawNodes             | draw vehicle path nodes<br>type: bool (cheat)<br>default: 0 (false)   |
| bot_drawObstacles         | Draw the bot's dynamic obstacles in the world<br>type: bool (cheat)<br>default: 0 (false)   |
| bot_drawRoutes            | Draw the bot's routes.<br>type: bool (cheat)<br>default: 0 (false)  |
| bot_enable                | 0 = bots will not be loaded in the game. 1 = bots are loaded.<br>type: bool<br>default: 1 (true)  |
| bot_fakePing              | Have the bots display fake pings in the scoreboard<br>type: int   |
| bot_followMe              | Have the bots follow you in debug mode<br>type: bool (cheat)<br>default: 0 (false)  |
| bot_hud                   | Print out the bots debug info onto the players HUD. Set to the clientNum of the bot to debug. 0 = disable. Make sure bot_debug is 1.<br>type: int (cheat)   |
| bot_ignoreEnemies         | If set to 1, bots will ignore all enemies. Useful for debugging bot behavior<br>type: int (cheat)   |
| bot_ignoreGoals           | If set to 1, bots will ignore all map objectives. Useful for debugging bot behavior<br>type: int (cheat)  |
| bot_knifeOnly             | goofy mode where the bots only use their knives in combat.<br>type: bool (cheat)<br>default: 0 (false)  |
| bot_minClients            | Keep a minimum number of clients on the server with bots. -1 to disable<br>type: int  |
| bot_noChat                | 0 = bots chat, 1 = bots never chat<br>type: bool<br>default: 0 (false)  |
| bot_noRandomJump          | makes bots not randomly jump<br>type: bool (cheat)<br>default: 0 (false)  |

| <b>cvar</b>               | <b>Description</b>   |
|---------------------------|--|
| bot_noTapOut              | makes bots not want to ever tap out, for debug purposes<br>type: bool (cheat)<br>default: 0 (false)                            |
| bot_noTaunt               | 0 = bots taunt, 1 = bots never taunt<br>type: bool<br>default: 1 (true)  |
| bot_pause                 | Pause the bot's thinking - useful for screenshots/debugging/etc<br>type: bool (cheat)<br>default: 0 (false)                    |
| bot_showPath              | Show the path for the bot's client number. -1 = disable.<br>type: int (cheat)  |
| bot_sillyWarmup           | 0 = bots play the game like normal, 1 = bots shoot each other and act silly during warmup<br>type: bool<br>default: 1 (true)   |
| bot_skill                 | Sets the bot's default AI skill. 0 = EASY, 1 = MEDIUM, 2 = EXPERT, 3 = MASTER<br>type: int                                     |
| bot_skipThinkClient       | A debug only cvar that skips thinking for a particular bot with the client number entered. -1 = disabled.<br>type: int (cheat) |
| bot_suicide               | Causes all bots to suicide. Useful for debugging.<br>type: bool (cheat)<br>default: 0 (false)                                  |
| bot_testObstacleAvoidance | test obstacle avoidance<br>type: bool (cheat)<br>default: 0 (false)  |
| bot_threadFixedDelay      | force the bot AI to always trail bot_threadFrameDelay frames behind<br>type: bool<br>default: 0 (false)                        |
| bot_threadFrameDelay      | maximum number of game frames the bot AI can trail behind<br>type: int [0, 4]  |
| bot_threading             | enable running the bot AI in a separate thread<br>type: bool<br>default: 1 (true)  |
| bot_uiNumGDF              | The number of gdf bots to add to the server. -1 to disable<br>type: int  |
| bot_uiNumStrogg           | The number of strogg bots to add to the server. -1 to disable<br>type: int   |
| bot_uiSkill               | The overall skill the bots should play at in the game. 0 = EASY, 1 = MEDIUM, 2 = EXPERT, 3 = MASTER<br>type: int               |
| bot_useShotguns           | 0 = bots wont use shotguns/nailguns. 1 = bots will use shotguns/nailguns.<br>type: bool<br>default: 1 (true)                   |
| bot_useSpawnHosts         | 0 = strogg bots can't use spawn host bodies, 1 = bots can use spawnhosts<br>type: bool<br>default: 1 (true)                    |

| <b>cvar</b>          | <b>Description</b>  |
|----------------------|---|
| bot_useVehicles      | 0 = bots dont use vehicles, 1 = bots do use vehicles<br>type: bool (cheat)<br>default: 1 (true) |
| bse_debris           | disable effect debris<br>type: bool (cheat)<br>default: 1 (true)                                |
| bse_debug            | display debug info about effect<br>type: int (cheat)  |
| bse_detailLevel      | Detail setting for effects<br>type: float   |
| bse_effectsStats     | type: string (cheat)  |
| bse_enabled          | set to false to disable all effects<br>type: bool (cheat)<br>default: 1 (true)                  |
| bse_maxDecalDistance | Maximum distance from the viewer to spawn decals<br>type: float (cheat)                         |
| bse_physics          | disable effect physics<br>type: bool (cheat)<br>default: 1 (true)                               |
| bse_projectileEffect | this effect will replace projectile explosions<br>type: string (cheat)                          |
| bse_rateCost         | rate cost multiplier for spawned effects<br>type: float   |
| bse_rateLimit        | rate limit for spawned effects<br>type: float   |
| bse_render           | disable effect rendering<br>type: bool (cheat)<br>default: 1 (true)                             |
| bse_showBounds       | display debug bounding boxes effect<br>type: int (cheat)  |
| bse_simple           | simple versions of effects<br>type: bool<br>default: 0 (false)                                  |
| bse_singleEffect     | set to the name of the effect that is only played<br>type: string (cheat)                       |
| bse_speeds           | print bse frame statistics<br>type: int (cheat)   |
| cm_backFaceCull      | cull back facing polygons<br>type: bool (cheat)<br>default: 0 (false)                           |
| cm_debugCollision    | debug the collision detection<br>type: bool (cheat)<br>default: 0 (false)                       |
| cm_debugRotation     | debug rotation failure<br>type: bool (cheat)<br>default: 1 (true)                               |
| cm_debugSetParms     | debug set parameters<br>type: bool (cheat)<br>default: 1 (true)                                 |



| <b>cvar</b>                 | <b>Description</b>   |
|-----------------------------|--|
| cm_debugTranslation         | debug translation failure<br>type: bool (cheat)<br>default: 1 (true)   |
| cm_drawColor                | color used to draw the collision models<br>type: string (cheat)  |
| cm_drawFilled               | draw filled polygons<br>type: bool (cheat)<br>default: 0 (false)   |
| cm_drawIndexScale           | scale of primitive indices<br>type: float (cheat)  |
| cm_drawIndices              | draw primitive indices<br>type: bool (cheat)<br>default: 0 (false)   |
| cm_drawInternal             | draw internal edges green<br>type: bool (cheat)<br>default: 1 (true)   |
| cm_drawMask                 | collision mask<br>type: string [none, solid, opaque, water, playerclip, moveableclip, ikclip, slidemover, body, corpse, trigger, vehicleclip, rendermodel, projectile, explosion, monster, forcefield] (cheat) |
| cm_drawNormals              | draw polygon and edge normals<br>type: float (cheat)   |
| cm_drawTraces               | draw polygon and edge normals<br>type: bool (cheat)<br>default: 0 (false)  |
| cm_loadRenderModels         | allow loading of render models for collision at run time<br>type: bool (cheat)<br>default: 1 (true)  |
| cm_showCompiledModelStats   | show stats for all compiled collision models<br>type: bool (cheat)<br>default: 0 (false)   |
| cm_showConversionWarnings   | sets whether to show warnings when converting models to trace models<br>type: bool (cheat)<br>default: 0 (false)   |
| cm_storeCollisionModelStats | any models converted will be stored for output later using the command dumpCollisionModelStats<br>type: bool (cheat)<br>default: 0 (false)   |
| cm_testAngle                | type: float (cheat)  |
| cm_testBox                  | type: string (cheat)   |
| cm_testBoxRotation          | type: string (cheat)   |
| cm_testCollision            | type: bool (cheat)<br>default: 0 (false)   |
| cm_testLength               | type: float (cheat)  |
| cm_testModel                | type: int (cheat)  |
| cm_testOrigin               | type: string (cheat)   |
| cm_testRadius               | type: float (cheat)  |

| <b>cvar</b>                     | <b>Description</b>  |
|---------------------------------|---|
| cm_testRandomMany               | type: bool (cheat)<br>default: 0 (false)  |
| cm_testReset                    | type: bool (cheat)<br>default: 0 (false)  |
| cm_testRotation                 | type: bool (cheat)<br>default: 1 (true)   |
| cm_testTimes                    | type: int (cheat)   |
| cm_testTranslation              | type: bool (cheat)<br>default: 1 (true)   |
| cm_testWalk                     | type: bool (cheat)<br>default: 1 (true)   |
| cm_writeCompiledCollisionModels | write out generated collision models to disk<br>type: bool (cheat)<br>default: 0 (false)  |
| com_allowConsole                | allow toggling console with the tilde key<br>type: bool<br>default: 0 (false)   |
| com_asyncInput                  | sample input from the async thread<br>type: bool (cheat)<br>default: 0 (false)  |
| com_asyncSound                  | 0: mix sound inline, 1: memory mapped async mix, 2: callback mixing, 3: write async mix<br>type: int [0, 1] (cheat)               |
| com_aviDemoHeight               | type: int (cheat)   |
| com_aviDemoSamples              | type: int (cheat)   |
| com_aviDemoTics                 | type: int (cheat)   |
| com_aviDemoWidth                | type: int (cheat)   |
| com_compressDemos               | type: bool (cheat)<br>default: 0 (false)  |
| com_fixedTic                    | type: bool (cheat)<br>default: 0 (false)  |
| com_forceGenericSIMD            | force generic platform independent SIMD<br>type: bool<br>default: 0 (false)   |
| com_gpuSpec                     | hardware classification, -1 = not detected, 0 = low quality, 1 = medium quality, 2 = high quality, 3 = ultra quality<br>type: int |
| com_journal                     | 1 = record journal, 2 = play back journal<br>type: int [0, 2] (init)  |
| com_logDemos                    | type: bool (cheat)<br>default: 0 (false)  |
| com_machineSpec                 | hardware classification, -1 = not detected, 0 = low quality, 1 = medium quality, 2 = high quality<br>type: int                    |
| com_makingBuild                 | 1 when making a build<br>type: bool (cheat)<br>default: 0 (false)   |

| <b>cvar</b>           | <b>Description</b>   |
|-----------------------|--|
| com_makingRC          | 1 when making a release candidate<br>type: bool (cheat)<br>default: 0 (false)          |
| com_minTics           | type: int (cheat)  |
| com_preciseTic        | run one game tick every async thread update<br>type: bool (cheat)<br>default: 1 (true) |
| com_preloadDemos      | type: bool (cheat)<br>default: 1 (true)  |
| com_purgeAll          | purge everything between level loads<br>type: bool<br>default: 0 (false)               |
| com_showAngles        | type: bool (cheat)<br>default: 0 (false)   |
| com_showBPS           | show bot think frames per second<br>type: bool<br>default: 0 (false)                   |
| com_showDemo          | type: bool (cheat)<br>default: 0 (false)   |
| com_showFPS           | show frames rendered per second<br>type: bool<br>default: 0 (false)                    |
| com_showMemoryUsage   | show total and per frame memory usage<br>type: bool<br>default: 0 (false)              |
| com_showSoundDecoders | show sound decoders<br>type: bool<br>default: 0 (false)                                |
| com_showTics          | type: bool (cheat)<br>default: 0 (false)   |
| com_showTPS           | show mega tiles loaded per second<br>type: bool<br>default: 0 (false)                  |
| com_skipGameDraw      | type: bool (cheat)<br>default: 0 (false)   |
| com_skipRenderer      | skip the renderer completely<br>type: bool (cheat)<br>default: 0 (false)               |
| com_skipWarnings      | skip all warnings<br>type: bool<br>default: 0 (false)                                  |
| com_speeds            | show engine timings<br>type: bool<br>default: 0 (false)                                |
| com_timestampPrints   | print time with each console print, 1 = msec, 2 = sec<br>type: string (cheat)          |
| com_unlock_avgFrames  | average out render times over n frames for extra renders management<br>type: int       |

| <b>cvar</b>             | <b>Description</b>  |
|-------------------------|---|
| com_unlock_maxFPS       | maximum FPS unlock will try to reach (only works in timing methods 1 & 2) - zero = no limit<br>type: int  |
| com_unlock_safetyMargin | percentage safety margin under which not to trigger an extra render<br>type: float  |
| com_unlock_timingMethod | method of timing interpolated/extrapolated frames: 0 = whenever they fit; 1 = on boundaries of framerate multiples of 30; 2 = like 1, but using total previous everything time to estimate, not just render time<br>type: int |
| com_unlockFPS           | allow renderer to go faster than game ticks<br>type: bool<br>default: 1 (true)  |
| com_useBinaryDecls      | Load binary versions of declarations falling back to text if not found<br>type: bool (cheat)<br>default: 1 (true)   |
| com_useFastVidRestart   | Allow use of fast vid restarts<br>type: bool<br>default: 0 (false)  |
| com_videoRam            | holds the last amount of detected video ram<br>type: int  |
| com_wipeSeconds         | type: float (cheat)   |
| com_writeBinaryDecls    | Write binary versions of declarations<br>type: bool (cheat)<br>default: 0 (false)   |
| con_noPrint             | print on the console but not onscreen when console is pulled up<br>type: bool<br>default: 1 (true)  |
| con_notifyTime          | time messages are displayed onscreen when console is pulled up<br>type: float   |
| con_speed               | speed at which the console moves up and down<br>type: float   |
| decl_show               | set to 1 to print parses, 2 to also print references, 3 only prints out of level load, 4 only prints out of level load but also prints references<br>type: int [0, 4] (cheat)   |
| decl_usageLog           | creates a log of all declarations touched<br>type: bool (cheat)<br>default: 0 (false)   |
| demo_debug              | debug demo replay system<br>type: bool<br>default: 0 (false)  |
| demo_noclip             | noclip through a demo<br>type: bool (cheat)<br>default: 0 (false)   |
| demo_prediction         | prediction when playing back a server demo in milliseconds<br>type: int   |

| <b>cvar</b>                  | <b>Description</b>   |
|------------------------------|--|
| demo_scale                   | speed scaling of demo replays<br>type: float [0.01, 100]   |
| demo_snapshotDelay           | delay between snapshots for server side demo recordings<br>type: int                                   |
| developer                    | developer mode<br>type: bool<br>default: 0 (false)   |
| fs_basepath                  | type: string (init)  |
| fs_caseSensitiveOS           | type: bool (cheat)<br>default: 0 (false)   |
| fs_cdpath                    | type: string (init)  |
| fs_copyfiles                 | type: int [0, 5] (init)  |
| fs_debug                     | type: int [0, 2] (cheat)   |
| fs_devpath                   | type: string (init)  |
| fs_game                      | mod path<br>type: string (init)  |
| fs_game_base                 | alternate mod path, searched after the main fs_game path,<br>before the basedir<br>type: string (init) |
| fs_restrict                  | type: bool (init)<br>default: 0 (false)  |
| fs_savepath                  | type: string (init)  |
| fs_searchAddons              | search all addon pk4s ( disables addon functionality )<br>type: bool (cheat)<br>default: 0 (false)     |
| fs_toolsPath                 | quoted paths that are searched for tools<br>type: string (cheat)                                       |
| fs_userpath                  | type: string (init)  |
| g_allowComplaint_charge      | Allow complaints for teamkills with charges<br>type: bool<br>default: 0 (false)                        |
| g_allowComplaint_explosives  | Allow complaints for explosive weapons and items<br>type: bool<br>default: 1 (true)                    |
| g_allowComplaint_firesupport | Allow complaints for teamkills with fire support<br>type: bool<br>default: 1 (true)                    |
| g_allowLateJoin              | Enable/disable players joining a match in progress<br>type: bool<br>default: 1 (true)                  |
| g_autoFireTeam               | Prompt to join a fireteam when switching to a new team.<br>type: bool<br>default: 0 (false)            |
| g_banner_1                   | banner message 1<br>type: string   |
| g_banner_10                  | banner message 10<br>type: string  |
| g_banner_11                  | banner message 11<br>type: string  |

| <b>cvar</b>              | <b>Description</b>   |
|--------------------------|--|
| g_banner_12              | banner message 12<br>type: string  |
| g_banner_13              | banner message 13<br>type: string  |
| g_banner_14              | banner message 14<br>type: string  |
| g_banner_15              | banner message 15<br>type: string  |
| g_banner_16              | banner message 16<br>type: string  |
| g_banner_2               | banner message 2<br>type: string   |
| g_banner_3               | banner message 3<br>type: string   |
| g_banner_4               | banner message 4<br>type: string   |
| g_banner_5               | banner message 5<br>type: string   |
| g_banner_6               | banner message 6<br>type: string   |
| g_banner_7               | banner message 7<br>type: string   |
| g_banner_8               | banner message 8<br>type: string   |
| g_banner_9               | banner message 9<br>type: string   |
| g_banner_delay           | delay between banner messages<br>type: int                                     |
| g_banner_loopdelay       | delay before banner messages repeat, 0 = off<br>type: int                      |
| g_cacheDictionaryMedia   | Precache all media from entity dictionaries<br>type: bool<br>default: 1 (true) |
| g_chatDefaultColor       | RGBA value for normal chat prints<br>type: string                              |
| g_chatFireTeamColor      | RGBA value for fire team chat prints<br>type: string                           |
| g_chatLineTimeout        | number of seconds that each chat line stays in the history<br>type: float      |
| g_chatTeamColor          | RGBA value for team chat prints<br>type: string                                |
| g_cheapDecalsMaxDistance | max distance decals are created<br>type: string                                |
| g_collisionModelMask     | type: int (cheat)  |
| g_commandMapZoom         | command map zoom level<br>type: float [0.125, 0.75]                            |
| g_commandMapZoomStep     | percent to increase/decrease command map zoom by<br>type: float                |

| <b>cvar</b>                 | <b>Description</b>  |
|-----------------------------|---|
| g_compiledScriptSafety      | enables extra safety checks in exported scripts<br>type: bool (cheat)<br>default: 1 (true)                                      |
| g_complaintGUIDLimit        | Total unique complaints at which a player will be kicked<br>type: int   |
| g_complaintLimit            | Total complaints at which a player will be kicked<br>type: int  |
| g_damageIndicatorAlphaScale | alpha of the damage indicators<br>type: float   |
| g_damageIndicatorColor      | color of the damage indicators<br>type: string  |
| g_damageIndicatorFadeTime   | number of seconds that a damage indicator stays visible<br>type: float  |
| g_damageIndicatorHeight     | height of the damage indicators<br>type: float  |
| g_damageIndicatorWidth      | width of the damage indicators<br>type: float   |
| g_debugAnim                 | displays information on which animations are playing on the specified entity number. set to -1 to disable.<br>type: int (cheat) |
| g_debugAnimLookups          | prints successful animation lookups<br>type: int (cheat)  |
| g_debugAnimStance           | displays information on which stances are set on the specified entity number. set to -1 to disable.<br>type: int (cheat)        |
| g_debugBounds               | checks for models with bounds > 2048<br>type: bool (cheat)<br>default: 0 (false)  |
| g_debugCinematic            | type: bool (cheat)<br>default: 0 (false)  |
| g_debugDamage               | type: bool (cheat)<br>default: 0 (false)  |
| g_debugFootsteps            | prints which surfacetype the player is walking on<br>type: bool (cheat)<br>default: 0 (false)                                   |
| g_debugForceFields          | type: bool (cheat)<br>default: 0 (false)  |
| g_debugFrameCommands        | Prints out frame commands as they are called<br>type: bool (cheat)<br>default: 0 (false)  |
| g_debugFrameCommandsFilter  | Filter the type of framecommands<br>type: string (cheat)  |
| g_debugGUI                  | Show GUI window outlines<br>type: bool (cheat)<br>default: 0 (false)  |
| g_debugGUIEvents            | Show the results of events<br>type: int (cheat)   |
| g_debugGUIRenderWorld       | Output information for GUI-based renderWorlds<br>type: bool (cheat)<br>default: 0 (false)                                       |

| <b>cvar</b>                    | <b>Description</b>  |
|--------------------------------|---|
| g_debugGUITextRect             | Show windows' text rectangle outlines<br>type: bool (cheat)<br>default: 0 (false)             |
| g_debugGUITextScale            | Size that the debug GUI info font is drawn in.<br>type: float (cheat)                         |
| g_debugLocations               | type: bool (cheat)<br>default: 0 (false)  |
| g_debugMask                    | debugs a deployment mask<br>type: string (cheat)  |
| g_debugNetworkWrite            | type: bool (cheat)<br>default: 0 (false)  |
| g_debugPlayerList              | fills UI lists with fake players<br>type: int (cheat)   |
| g_debugProficiency             | type: bool (cheat)<br>default: 0 (false)  |
| g_debugScript                  | type: bool (cheat)<br>default: 0 (false)  |
| g_debugSpecatorList            | fills the spectator list with fake players<br>type: int (cheat)                               |
| g_debugVehicleDriveForces      | show info about wheeled drive forces<br>type: bool (cheat)<br>default: 0 (false)              |
| g_debugVehicleFeedback         | show info about wheeled suspension feedback<br>type: bool (cheat)<br>default: 0 (false)       |
| g_debugVehicleFrictionFeedback | show info about wheeled surface friction feedback<br>type: bool (cheat)<br>default: 0 (false) |
| g_debugVehicleHoverPads        | show info about hoverpads<br>type: bool (cheat)<br>default: 0 (false)                         |
| g_debugVehicleHurtZones        | show info about the hurtZone component<br>type: bool (cheat)<br>default: 0 (false)            |
| g_debugVehiclePseudoHover      | show info about the pseudoHover component<br>type: bool (cheat)<br>default: 0 (false)         |
| g_debugVehicleWheelForces      | show info about wheel forces<br>type: bool (cheat)<br>default: 0 (false)                      |
| g_debugWakes                   | Debug the vehicle wakes<br>type: bool (cheat)<br>default: 0 (false)                           |
| g_debugWeapon                  | type: bool (cheat)<br>default: 0 (false)  |
| g_debugWeaponSpread            | displays the current spread value for the weapon<br>type: bool (cheat)<br>default: 0 (false)  |



| <b>cvar</b>                 | <b>Description</b>  |
|-----------------------------|---|
| g_debugWorldTasks           | draws a sphere around ranged based tasks<br>type: bool (cheat)<br>default: 0 (false)  |
| g_decals                    | show decals such as bullet holes<br>type: bool<br>default: 1 (true)   |
| g_demoAnalyzationSectorSize | sector size for stat generation<br>type: int (cheat)  |
| g_demoAnalyze               | analyze demo during playback<br>type: bool (cheat)<br>default: 0 (false)  |
| g_demoOutputMDF             | output entity keyframe data from demo<br>type: int [0, 2] (cheat)   |
| g_disableFootsteps          | enable/disable footsteps<br>type: bool (cheat)<br>default: 0 (false)  |
| g_disableGlobalAudio        | disable global VOIP communication<br>type: bool<br>default: 0 (false)   |
| g_disableGlobalChat         | disable global text communication<br>type: bool<br>default: 0 (false)   |
| g_disableTransportDebris    | type: bool (cheat)<br>default: 0 (false)  |
| g_disableVehicleSpawns      | disables vehicles spawning from construction pads<br>type: string (cheat)   |
| g_disasm                    | disassemble script into base/script/disasm.txt on the local drive when script is compiled<br>type: bool (cheat)<br>default: 0 (false)               |
| g_dragDamping               | type: float (cheat)   |
| g_dragEntity                | allows dragging physics objects around by placing the crosshair over them and holding the fire button<br>type: bool (cheat)<br>default: 0 (false)   |
| g_dragMaxforce              | type: float (cheat)   |
| g_dragShowSelection         | type: bool (cheat)<br>default: 0 (false)  |
| g_drawContacts              | draw physics object contacts<br>type: bool (cheat)<br>default: 0 (false)  |
| g_drawPlayerDamage          | Draws numbers above the player's head every time they take damage ( Must be enabled on the server too )<br>type: bool (cheat)<br>default: 0 (false) |
| g_drawPlayerIcons           | Enables/Disables player icons<br>type: bool<br>default: 1 (true)  |
| g_drawRouteConstraints      | draws lines showing route constraints<br>type: int (cheat)  |

| cvar                  | Description   |
|-----------------------|---|
| g_editEntityMode      | 0 = off<br>1 = lights<br>2 = sounds<br>3 = articulated figures<br>4 = particle systems<br>5 = monsters<br>6 = entity names<br>7 = entity models<br>type: int [0, 7] (cheat) |
| g_enemyColor          | color of enemy units<br>type: string  |
| g_execMapConfigs      | Execute map cfg with same name<br>type: bool<br>default: 0 (false)  |
| g_exportMask          | type: string (cheat)  |
| g_forceClear          | forces clearing of color buffer on main game draw (faster)<br>type: bool (cheat)<br>default: 1 (true)   |
| g_fov                 | type: int   |
| g_frametime           | displays timing information for each game frame<br>type: bool (cheat)<br>default: 0 (false)   |
| g_friendlyColor       | color of friendly units<br>type: string   |
| g_gameReviewPause     | Time (in minutes) for scores review time<br>type: float   |
| g_gameReviewReadyWait | wait for players to ready up before going to the next map<br>type: bool<br>default: 0 (false)   |
| g_gravity             | type: float (cheat)   |
| g_guiSpeeds           | Show GUI speeds<br>type: bool (cheat)<br>default: 0 (false)   |
| g_gunX                | type: float (cheat)   |
| g_gunY                | type: float (cheat)   |
| g_gunZ                | type: float (cheat)   |
| g_hitBeep             | play hit beep sound when you inflict damage.<br>0 = do nothing<br>1 = beep/flash cross-hair<br>2 = beep<br>3 = flash cross-hair<br>type: int                                |
| g_kickAmplitude       | type: float (cheat)   |
| g_kickBanLength       | length of time a kicked player will be banned for<br>type: float  |
| g_kickTime            | type: float (cheat)   |
| g_knockback           | type: int (cheat)   |
| g_logObjectives       | log objective completion info<br>type: bool<br>default: 1 (true)  |

| <b>cvar</b>                   | <b>Description</b>   |
|-------------------------------|--|
| g_logProficiency              | log proficiency data<br>type: bool<br>default: 1 (true)  |
| g_maxPlayerWarnings           | maximum warnings before player is kicked<br>type: int  |
| g_maxShowDistance             | type: float (cheat)  |
| g_maxSpectateTime             | maximum length of time a player may spectate for<br>type: float  |
| g_maxTransportDebrisExtraHigh | The maximum number of pieces of extra high priority (really large) debris. -1 means no limit.<br>type: int |
| g_maxTransportDebrisHigh      | The maximum number of pieces of high priority (large) debris. -1 means no limit.<br>type: int              |
| g_maxTransportDebrisLow       | The maximum number of pieces of low priority (small) debris. -1 means no limit.<br>type: int               |
| g_maxTransportDebrisMedium    | The maximum number of pieces of medium priority (middling) debris. -1 means no limit.<br>type: int         |
| g_maxVoiceChats               | maximum number of voice chats a player may do in a period of time<br>type: int                             |
| g_maxVoiceChatsOver           | time over which the maximum number of voice chat limit is applied<br>type: int                             |
| g_muteSpecs                   | Send all spectator global chat to team chat<br>type: bool<br>default: 0 (false)                            |
| g_neutralColor                | color of neutral units<br>type: string   |
| g_nextMap                     | commands to execute when the current map/campaign ends<br>type: string                                     |
| g_noBotSpectate               | disables the ability to spectate bots<br>type: bool<br>default: 1 (true)                                   |
| g_noProficiency               | enable/disable XP<br>type: bool<br>default: 0 (false)  |
| g_noRouteConstraintKick       | enables/disables players being kicked for deviating from routes<br>type: bool<br>default: 0 (false)        |
| g_noRouteMaskDestruction      | enables/disables the mcp being destroyed when driven outside the mask<br>type: bool<br>default: 0 (false)  |
| g_noVehicleDecay              | enables / disables vehicle decay<br>type: bool (cheat)<br>default: 0 (false)                               |
| g_password                    | game password<br>type: string  |

| <b>cvar</b>                    | <b>Description</b>  |
|--------------------------------|---|
| g_playerArrowIconSize          | Size of the screen space player arrow icons<br>type: float  |
| g_playerIconAlphaScale         | alpha to apply to world-based objective icons<br>type: float  |
| g_playerIconSize               | Size of the screen space player icons<br>type: float  |
| g_playerPushForce              | force players can be pushed by other players<br>type: float (cheat)   |
| g_playTooltipSound             | Play the sound attached to a tooltip when it is shown<br>type: bool<br>default: 1 (true)                    |
| g_privatePassword              | game password for private slots<br>type: string   |
| g_radialMenuMouseInput         | 0 - no mouse input<br>1 - mouse input, no view movement<br>2 - mouse input, view movement<br>type: int      |
| g_radialMenuMouseSensitivity   | Mouse input scale<br>type: float  |
| g_radialMenuStyle              | Sets the style of the quick chat menu: 0 = radial, 1 = vertical<br>type: int                                |
| g_radialMenuUseNumberShortcuts | Use numbers instead of alpha-numeric shortcuts<br>type: bool<br>default: 1 (true)                           |
| g_removeStaticEntities         | Remove non-dynamic entities on map spawn when they aren't needed<br>type: bool (cheat)<br>default: 1 (true) |
| g_rotateCommandMap             | Rotate the command map around the player<br>type: bool<br>default: 1 (true)                                 |
| g_showActiveEntities           | draws boxes around thinking entities.<br>type: bool (cheat)<br>default: 0 (false)                           |
| g_showAreaClipSectors          | type: float (cheat)   |
| g_showcamerainfo               | displays the current frame # for the camera when playing cinematics<br>type: string (cheat)                 |
| g_showChatLocation             | show/hide locations in chat text<br>type: bool (cheat)<br>default: 1 (true)                                 |
| g_showClipSectorFilter         | type: string (cheat)  |
| g_showClipSectors              | type: bool (cheat)<br>default: 0 (false)  |
| g_showCollisionModels          | type: bool (cheat)<br>default: 0 (false)  |
| g_showCollisionTraces          | type: bool (cheat)<br>default: 0 (false)  |
| g_showCollisionWorld           | type: int (cheat)   |

| <b>cvar</b>             | <b>Description</b>   |
|-------------------------|--|
| g_showCommandMapNames   | Show/hide location text on the commandmap<br>type: bool (cheat)<br>default: 0 (false)  |
| g_showCompileStats      | sets whether to show stats at the end of compilation or not<br>type: bool (cheat)<br>default: 0 (false)                                  |
| g_showCrosshairInfo     | shows information about the entity under your crosshair<br>type: int (cheat)   |
| g_showDemoHud           | draw the demo hud gui<br>type: bool (cheat)<br>default: 0 (false)  |
| g_showDemoView          | show player's calculated view when paused instead of free-fly cam<br>type: bool (cheat)<br>default: 0 (false)                            |
| g_showEntityInfo        | type: bool (cheat)<br>default: 0 (false)   |
| g_showEntityInfoPrint   | type: bool (cheat)<br>default: 0 (false)   |
| g_showFireTeamLocation  | show/hide locations in fireteam box text<br>type: bool<br>default: 0 (false)   |
| g_showHud               | draw the hud gui<br>type: bool (cheat)<br>default: 1 (true)  |
| g_showLocationWayPoints | Show/hide location waypoints in the world<br>type: int (cheat)   |
| g_showPlayerShadow      | enables shadow of player model<br>type: bool<br>default: 1 (true)  |
| g_showPlayerSpeed       | displays player movement speed<br>type: bool<br>default: 0 (false)   |
| g_showPVS               | type: int [0, 2] (cheat)   |
| g_showRenderModelBounds | type: bool (cheat)<br>default: 0 (false)   |
| g_showTargets           | draws entities and their targets. hidden entities are drawn grey.<br>type: bool (cheat)<br>default: 0 (false)                            |
| g_showTestModelFrame    | displays the current animation and frame # for testmodels<br>type: bool (cheat)<br>default: 0 (false)                                    |
| g_showTriggers          | draws trigger entities (orange) and their targets (green). disabled triggers are drawn grey.<br>type: bool (cheat)<br>default: 0 (false) |
| g_showVehiclePathNodes  | type: int (cheat)  |
| g_showWayPoints         | show or hide world-based objective icons<br>type: bool<br>default: 1 (true)  |

| <b>cvar</b>                   | <b>Description</b>  |
|-------------------------------|---|
| g_skipDeployChecks            | Skips deployment territory checks, etc<br>type: bool (cheat)<br>default: 0 (false)  |
| g_skipIntro                   | skip the opening intro movie<br>type: bool (read only)<br>default: 1 (true)   |
| g_skipLocalizedPrecipitation  | Enable/disable precipitation effects<br>type: bool (cheat)<br>default: 0 (false)  |
| g_skipPostProcess             | draw the post process gui<br>type: bool (cheat)<br>default: 0 (false)   |
| g_skipPrecipitation           | Enable/disable precipitation effects<br>type: bool (cheat)<br>default: 0 (false)  |
| g_skipVehicleAccelFeedback    | skip acceleration effects on wheeled suspensions<br>type: bool (cheat)<br>default: 0 (false)  |
| g_skipVehicleFrictionFeedback | ignore the effects of surface friction<br>type: bool (cheat)<br>default: 0 (false)  |
| g_skipVehicleTurnFeedback     | skip turn ducking effects on wheeled suspensions<br>type: bool (cheat)<br>default: 0 (false)  |
| g_skipViewEffects             | skip damage and other view effects<br>type: bool (cheat)<br>default: 0 (false)  |
| g_skipWeaponSwitchAnimations  | If 1, players won't play their weapon switching animations.<br>type: bool (cheat)<br>default: 1 (true)  |
| g_spectateLerpScale           | Controls view smoothing for spectators<br>type: float [0.2, 1]  |
| g_stopWatchMode               | stopwatch mode, 0 = ABBA, 1 = ABAB<br>type: int (cheat)   |
| g_teamSwitchDelay             | Delay (in seconds) before player can change teams again<br>type: int  |
| g_testLadders                 | type: bool (cheat)<br>default: 0 (false)  |
| g_testModelAnimate            | test model animation,<br>0 = cycle anim with origin reset<br>1 = cycle anim with fixed origin<br>2 = cycle anim with continuous origin<br>3 = frame by frame with continuous origin<br>4 = play anim once<br>type: int [0, 4] (cheat) |
| g_testModelBlend              | number of frames to blend<br>type: int (cheat)  |
| g_testModelRotate             | test model rotation speed<br>type: string (cheat)   |
| g_testParticle                | test particle visualization, set by the particle editor<br>type: int (cheat)  |

| <b>cvar</b>                      | <b>Description</b>  |
|----------------------------------|---|
| g_testParticleName               | name of the particle being tested by the particle editor<br>type: string (cheat)  |
| g_testPostProcess                | name of material to draw over screen<br>type: string (cheat)  |
| g_testSpectator                  | type: int (cheat)   |
| g_testViewSkin                   | name of skin to use for the view<br>type: string (cheat)  |
| g_transportDebrisExtraHighCutoff | Beyond this distance from the viewpoint extra high priority debris will not be spawned. -1 means no limit.<br>type: int |
| g_transportDebrisHighCutoff      | Beyond this distance from the viewpoint high priority debris will not be spawned. -1 means no limit.<br>type: int       |
| g_transportDebrisLowCutoff       | Beyond this distance from the viewpoint low priority debris will not be spawned. -1 means no limit.<br>type: int        |
| g_transportDebrisMediumCutoff    | Beyond this distance from the viewpoint medium priority debris will not be spawned. -1 means no limit.<br>type: int     |
| g_unlock_interpolateMoving       | interpolate moving objects in fps unlock mode<br>type: bool<br>default: 1 (true)  |
| g_unlock_updateAngles            | update view angles in fps unlock mode<br>type: bool<br>default: 1 (true)  |
| g_unlock_updateViewpos           | update view origin in fps unlock mode<br>type: bool<br>default: 1 (true)  |
| g_unlock_viewStyle               | 0: extrapolate view origin, 1: interpolate view origin<br>type: int   |
| g_useCompiledScript              | enable/disable native compiled scripts<br>type: bool<br>default: 1 (true)   |
| g_useSimpleStats                 | only look up local server stats<br>type: bool (cheat)<br>default: 0 (false)   |
| g_useTraceCollection             | Use optimized trace collections<br>type: bool (cheat)<br>default: 1 (true)  |
| g_vehicleForce                   | type: float (cheat)   |
| g_vehicleSteerKeyScale           | The scale of the wheeled vehicle steering keys - 1 is standard, 2 is twice as fast, etc<br>type: float                  |
| g_vehicleSuspensionDamping       | type: float (cheat)   |
| g_vehicleSuspensionDown          | type: float (cheat)   |
| g_vehicleSuspensionKCompress     | type: float (cheat)   |
| g_vehicleSuspensionUp            | type: float (cheat)   |
| g_vehicleTireFriction            | type: float (cheat)   |
| g_vehicleVelocity                | type: float (cheat)   |

| <b>cvar</b>                  | <b>Description</b>   |
|------------------------------|--|
| g_vehicleWheelTracesPerFrame | What fraction of the wheels are updated per frame<br>type: float (cheat)   |
| g_version                    | game version<br>type: string (read only)   |
| g_volumeMusic_dB             | music volume in dB<br>type: float  |
| g_votePassPercentage         | Percentage of yes votes required for a vote to pass<br>type: float [0, 100]                                      |
| g_voteWait                   | Delay (in minutes) before player may perform a callvote again<br>type: float                                     |
| g_walkerTraceDistance        | distance to check for space for the walker to move<br>type: float  |
| g_warmup                     | Length (in minutes) of warmup period<br>type: float  |
| g_warmupDamage               | Enable/disable players taking damage during warmup<br>type: bool<br>default: 1 (true)                            |
| g_waypointAlphaScale         | alpha to apply to world-based objective icons<br>type: float   |
| g_waypointDistanceMax        | min distance at which to show max icon size<br>type: float   |
| g_waypointDistanceMin        | max distance at which to show min icon size<br>type: float   |
| g_waypointSizeMax            | max world-view icon size<br>type: float  |
| g_waypointSizeMin            | min world-view icon size<br>type: float  |
| g_weaponSwitchTimeout        | type: float  |
| g_xpSave                     | stores xp for disconnected players which will be given back if they reconnect<br>type: bool<br>default: 1 (true) |
| gamedate                     | type: string (read only)   |
| gamename                     | type: string (read only)   |
| gui_crosshairAlpha           | alpha of crosshair<br>type: float  |
| gui_crosshairColor           | RGB color tint for crosshair elements<br>type: string  |
| gui_crosshairDef             | name of def containing crosshair<br>type: string   |
| gui_crosshairGrenadeAlpha    | alpha of grenade timer components<br>type: float   |
| gui_crosshairKey             | name of crosshair key in def specified by gui_crosshairDef<br>type: string                                       |
| gui_crosshairSpreadAlpha     | alpha of spread components<br>type: float  |
| gui_crosshairSpreadScale     | amount to scale the spread indicator movement<br>type: float   |



| <b>cvar</b>             | <b>Description</b>  |
|-------------------------|---|
| gui_crosshairStatsAlpha | alpha of health/ammo/reload components<br>type: float   |
| gui_debugLayout         | Debug UI layout classes<br>type: bool (cheat)<br>default: 0 (false)   |
| gui_debugRadialMenus    | Show radial menu debugging info<br>type: bool (cheat)<br>default: 0 (false)   |
| gui_doubleClickTime     | Delay in seconds between considering two mouse clicks a double-click<br>type: float   |
| gui_notificationPause   | length of time between successive notifications, in seconds<br>type: float  |
| gui_notificationTime    | length of time a user notification is on screen, in seconds<br>type: float  |
| gui_scoreBoardSort      | 0 - group by XP, 1 - group by fireteam, then by XP<br>type: int   |
| gui_showTooltips        | Globally enable or disable tooltips.<br>type: bool<br>default: 1 (true)   |
| gui_tooltipDelay        | Delay in seconds before tooltips pop up.<br>type: float   |
| ik_debug                | show IK debug lines<br>type: bool (cheat)<br>default: 0 (false)   |
| ik_enable               | enable IK<br>type: bool (cheat)<br>default: 1 (true)  |
| image_anisotropy        | set the maximum texture anisotropy if available<br>type: string   |
| image_bumpPicMip        | Uses a miplevel X steps down<br>type: int [-4, 2]   |
| image_colorMipLevels    | development aid to see texture mip usage<br>type: bool (cheat)<br>default: 0 (false)  |
| image_detailPower       | Controls how fast the detail textures fade out (0 = normal mipmaps, 1 is falloff after the first level)<br>type: string   |
| image_diffusePicMip     | Uses a miplevel X steps down<br>type: int [-4, 2]   |
| image_editorPicMip      | type: int [-4, 1] (cheat)   |
| image_filter            | changes texture filtering on mipmapped images<br>type: string [GL_LINEAR_MIPMAP_NEAREST, GL_LINEAR_MIPMAP_LINEAR, GL_NEAREST, GL_LINEAR, GL_NEAREST_MIPMAP_NEAREST, GL_NEAREST_MIPMAP_LINEAR] |
| image_globalPicMip      | globally uses a miplevel X steps down<br>type: int [-4, 2]  |
| image_ignoreHighQuality | ignore high quality setting on materials<br>type: string  |

| <b>cvar</b>                      | <b>Description</b>  |
|----------------------------------|---|
| image_lodbias                    | change lod bias on mipmapped images<br>type: float [-1, 1]  |
| image_lodDistance                | Distance at wich images don't need full resolution<br>type: float (cheat)   |
| image_picMip                     | Uses a miplevel X steps down<br>type: int [-4, 2]   |
| image_picMipEnable               | Enable picmip<br>type: int  |
| image_roundDown                  | round bad sizes down to nearest power of two<br>type: bool<br>default: 1 (true)                                   |
| image_showBackgroundLoads        | 1 = print number of outstanding background loads<br>type: bool (cheat)<br>default: 0 (false)                      |
| image_skipUpload                 | used during the build process, will skip uploads<br>type: bool (cheat)<br>default: 0 (false)                      |
| image_specularPicMip             | Uses a miplevel X steps down<br>type: int [-4, 2]   |
| image_useAllFormats              | allow alpha/intensity/luminance/luminance+alpha<br>type: bool<br>default: 1 (true)                                |
| image_useBackgroundLoads         | 1 = enable background loading of images<br>type: bool<br>default: 1 (true)  |
| image_useCompression             | 0 = force everything to high quality<br>type: bool<br>default: 1 (true)   |
| image_useNormalCompression       | 2 = use rrgb compression for normal maps, 1 = use 256 color compression for normal maps if available<br>type: int |
| image_useOfflineCompression      | write a batch file for offline compression of DDS files<br>type: bool (cheat)<br>default: 0 (false)               |
| image_usePrecompressedTextures   | use .dds files if present<br>type: string   |
| image_writeNormalTGA             | write .tgas of the final normal maps for debugging<br>type: bool (cheat)<br>default: 0 (false)                    |
| image_writeNormalTGAPalletized   | write .tgas of the final palletized normal maps for debugging<br>type: bool (cheat)<br>default: 0 (false)         |
| image_writePrecompressedTextures | write .dds files if necessary<br>type: bool (cheat)<br>default: 0 (false)   |
| image_writeTGA                   | write .tgas of the non normal maps for debugging<br>type: bool (cheat)<br>default: 0 (false)                      |
| in_anglespeedkey                 | angle change scale when holding down _speed button<br>type: float   |

| <b>cvar</b>              | <b>Description</b>   |
|--------------------------|--|
| in_car_pitch_axis        | which controller axis is used<br>type: int                                     |
| in_car_pitch_deadZone    | specifies how far large the dead-zone is on the controller axis<br>type: float |
| in_car_pitch_invert      | inverts the axis<br>type: bool<br>default: 1 (true)                            |
| in_car_pitch_joy         | the joystick number used<br>type: int  |
| in_car_pitch_offset      | the step up the dead zone<br>type: float                                       |
| in_car_pitch_power       | the power of the curve after dead zone - ie acceleration<br>type: float        |
| in_car_pitch_speed       | speed of the controller input<br>type: float                                   |
| in_car_steering_axis     | which controller axis is used<br>type: int                                     |
| in_car_steering_deadZone | specifies how far large the dead-zone is on the controller axis<br>type: float |
| in_car_steering_invert   | inverts the axis<br>type: bool<br>default: 0 (false)                           |
| in_car_steering_joy      | the joystick number used<br>type: int  |
| in_car_steering_offset   | the step up the dead zone<br>type: float                                       |
| in_car_steering_power    | the power of the curve after dead zone - ie acceleration<br>type: float        |
| in_car_steering_speed    | speed of the controller input<br>type: float                                   |
| in_car_throttle_axis     | which controller axis is used<br>type: int                                     |
| in_car_throttle_deadZone | specifies how far large the dead-zone is on the controller axis<br>type: float |
| in_car_throttle_invert   | inverts the axis<br>type: bool<br>default: 1 (true)                            |
| in_car_throttle_joy      | the joystick number used<br>type: int  |
| in_car_throttle_offset   | the step up the dead zone<br>type: float                                       |
| in_car_throttle_power    | the power of the curve after dead zone - ie acceleration<br>type: float        |
| in_car_throttle_speed    | speed of the controller input<br>type: float                                   |
| in_car_yaw_axis          | which controller axis is used<br>type: int                                     |
| in_car_yaw_deadZone      | specifies how far large the dead-zone is on the controller axis<br>type: float |

| <b>cvar</b>               | <b>Description</b>  |
|---------------------------|---|
| in_car_yaw_invert         | inverts the axis<br>type: bool<br>default: 1 (true)                               |
| in_car_yaw_joy            | the joystick number used<br>type: int   |
| in_car_yaw_offset         | the step up the dead zone<br>type: float  |
| in_car_yaw_power          | the power of the curve after dead zone - ie acceleration<br>type: float           |
| in_car_yaw_speed          | speed of the controller input<br>type: float                                      |
| in_freeLook               | look around with mouse (reverse _mlook button)<br>type: bool<br>default: 1 (true) |
| in_heli_forward_axis      | which controller axis is used<br>type: int  |
| in_heli_forward_deadZone  | specifies how far large the dead-zone is on the controller axis<br>type: float    |
| in_heli_forward_invert    | inverts the axis<br>type: bool<br>default: 0 (false)                              |
| in_heli_forward_joy       | the joystick number used<br>type: int   |
| in_heli_forward_offset    | the step up the dead zone<br>type: float  |
| in_heli_forward_power     | the power of the curve after dead zone - ie acceleration<br>type: float           |
| in_heli_forward_speed     | speed of the controller input<br>type: float                                      |
| in_heli_side_axis         | which controller axis is used<br>type: int  |
| in_heli_side_deadZone     | specifies how far large the dead-zone is on the controller axis<br>type: float    |
| in_heli_side_invert       | inverts the axis<br>type: bool<br>default: 1 (true)                               |
| in_heli_side_joy          | the joystick number used<br>type: int   |
| in_heli_side_offset       | the step up the dead zone<br>type: float  |
| in_heli_side_power        | the power of the curve after dead zone - ie acceleration<br>type: float           |
| in_heli_side_speed        | speed of the controller input<br>type: float                                      |
| in_heli_throttle_axis     | which controller axis is used<br>type: int  |
| in_heli_throttle_deadZone | specifies how far large the dead-zone is on the controller axis<br>type: float    |

| <b>cvar</b>                   | <b>Description</b>   |
|-------------------------------|--|
| in_heli_throttle_invert       | inverts the axis<br>type: bool<br>default: 1 (true)                            |
| in_heli_throttle_joy          | the joystick number used<br>type: int  |
| in_heli_throttle_offset       | the step up the dead zone<br>type: float                                       |
| in_heli_throttle_power        | the power of the curve after dead zone - ie acceleration<br>type: float        |
| in_heli_throttle_speed        | speed of the controller input<br>type: float                                   |
| in_heli_yaw_axis              | which controller axis is used<br>type: int                                     |
| in_heli_yaw_deadZone          | specifies how far large the dead-zone is on the controller axis<br>type: float |
| in_heli_yaw_invert            | inverts the axis<br>type: bool<br>default: 0 (false)                           |
| in_heli_yaw_joy               | the joystick number used<br>type: int  |
| in_heli_yaw_offset            | the step up the dead zone<br>type: float                                       |
| in_heli_yaw_power             | the power of the curve after dead zone - ie acceleration<br>type: float        |
| in_heli_yaw_speed             | speed of the controller input<br>type: float                                   |
| in_hovertank_forward_axis     | which controller axis is used<br>type: int                                     |
| in_hovertank_forward_deadZone | specifies how far large the dead-zone is on the controller axis<br>type: float |
| in_hovertank_forward_invert   | inverts the axis<br>type: bool<br>default: 1 (true)                            |
| in_hovertank_forward_joy      | the joystick number used<br>type: int  |
| in_hovertank_forward_offset   | the step up the dead zone<br>type: float                                       |
| in_hovertank_forward_power    | the power of the curve after dead zone - ie acceleration<br>type: float        |
| in_hovertank_forward_speed    | speed of the controller input<br>type: float                                   |
| in_hovertank_pitch_axis       | which controller axis is used<br>type: int                                     |
| in_hovertank_pitch_deadZone   | specifies how far large the dead-zone is on the controller axis<br>type: float |
| in_hovertank_pitch_invert     | inverts the axis<br>type: bool<br>default: 1 (true)                            |
| in_hovertank_pitch_joy        | the joystick number used<br>type: int  |

| <b>cvar</b>                | <b>Description</b>   |
|----------------------------|--|
| in_hovertank_pitch_offset  | the step up the dead zone<br>type: float                                       |
| in_hovertank_pitch_power   | the power of the curve after dead zone - ie acceleration<br>type: float        |
| in_hovertank_pitch_speed   | speed of the controller input<br>type: float                                   |
| in_hovertank_side_axis     | which controller axis is used<br>type: int                                     |
| in_hovertank_side_deadZone | specifies how far large the dead-zone is on the controller axis<br>type: float |
| in_hovertank_side_invert   | inverts the axis<br>type: bool<br>default: 0 (false)                           |
| in_hovertank_side_joy      | the joystick number used<br>type: int  |
| in_hovertank_side_offset   | the step up the dead zone<br>type: float                                       |
| in_hovertank_side_power    | the power of the curve after dead zone - ie acceleration<br>type: float        |
| in_hovertank_side_speed    | speed of the controller input<br>type: float                                   |
| in_hovertank_turn_axis     | which controller axis is used<br>type: int                                     |
| in_hovertank_turn_deadZone | specifies how far large the dead-zone is on the controller axis<br>type: float |
| in_hovertank_turn_invert   | inverts the axis<br>type: bool<br>default: 0 (false)                           |
| in_hovertank_turn_joy      | the joystick number used<br>type: int  |
| in_hovertank_turn_offset   | the step up the dead zone<br>type: float                                       |
| in_hovertank_turn_power    | the power of the curve after dead zone - ie acceleration<br>type: float        |
| in_hovertank_turn_speed    | speed of the controller input<br>type: float                                   |
| in_hovertank_yaw_axis      | which controller axis is used<br>type: int                                     |
| in_hovertank_yaw_deadZone  | specifies how far large the dead-zone is on the controller axis<br>type: float |
| in_hovertank_yaw_invert    | inverts the axis<br>type: bool<br>default: 1 (true)                            |
| in_hovertank_yaw_joy       | the joystick number used<br>type: int  |
| in_hovertank_yaw_offset    | the step up the dead zone<br>type: float                                       |
| in_hovertank_yaw_power     | the power of the curve after dead zone - ie acceleration<br>type: float        |

| <b>cvar</b>                | <b>Description</b>   |
|----------------------------|--|
| in_hovertank_yaw_speed     | speed of the controller input<br>type: float   |
| in_joy1_device             | the hash of the controller device named joy1<br>type: int                              |
| in_joy2_device             | the hash of the controller device named joy2<br>type: int                              |
| in_joy3_device             | the hash of the controller device named joy3<br>type: int                              |
| in_joy4_device             | the hash of the controller device named joy4<br>type: int                              |
| in_mouse                   | enable mouse input<br>type: bool (cheat)<br>default: 1 (true)                          |
| in_pitchspeed              | pitch change speed when holding down look_lookUp or<br>_lookDown button<br>type: float |
| in_player_forward_axis     | which controller axis is used<br>type: int   |
| in_player_forward_deadZone | specifies how far large the dead-zone is on the controller axis<br>type: float         |
| in_player_forward_invert   | inverts the axis<br>type: bool<br>default: 1 (true)                                    |
| in_player_forward_joy      | the joystick number used<br>type: int  |
| in_player_forward_offset   | the step up the dead zone<br>type: float   |
| in_player_forward_power    | the power of the curve after dead zone - ie acceleration<br>type: float                |
| in_player_forward_speed    | speed of the controller input<br>type: float   |
| in_player_pitch_axis       | which controller axis is used<br>type: int   |
| in_player_pitch_deadZone   | specifies how far large the dead-zone is on the controller axis<br>type: float         |
| in_player_pitch_invert     | inverts the axis<br>type: bool<br>default: 1 (true)                                    |
| in_player_pitch_joy        | the joystick number used<br>type: int  |
| in_player_pitch_offset     | the step up the dead zone<br>type: float   |
| in_player_pitch_power      | the power of the curve after dead zone - ie acceleration<br>type: float                |
| in_player_pitch_speed      | speed of the controller input<br>type: float   |
| in_player_side_axis        | which controller axis is used<br>type: int   |
| in_player_side_deadZone    | specifies how far large the dead-zone is on the controller axis<br>type: float         |

| <b>cvar</b>            | <b>Description</b>   |
|------------------------|--|
| in_player_side_invert  | inverts the axis<br>type: bool<br>default: 0 (false)   |
| in_player_side_joy     | the joystick number used<br>type: int  |
| in_player_side_offset  | the step up the dead zone<br>type: float   |
| in_player_side_power   | the power of the curve after dead zone - ie acceleration<br>type: float  |
| in_player_side_speed   | speed of the controller input<br>type: float   |
| in_player_yaw_axis     | which controller axis is used<br>type: int   |
| in_player_yaw_deadZone | specifies how far large the dead-zone is on the controller axis<br>type: float   |
| in_player_yaw_invert   | inverts the axis<br>type: bool<br>default: 1 (true)  |
| in_player_yaw_joy      | the joystick number used<br>type: int  |
| in_player_yaw_offset   | the step up the dead zone<br>type: float   |
| in_player_yaw_power    | the power of the curve after dead zone - ie acceleration<br>type: float  |
| in_player_yaw_speed    | speed of the controller input<br>type: float   |
| in_toggleRun           | pressing _speed button toggles run on/off<br>type: bool<br>default: 0 (false)  |
| in_toggleSprint        | pressing _sprint button toggles run on/off<br>type: bool<br>default: 0 (false)   |
| in_yawspeed            | yaw change speed when holding down _left or _right button<br>type: float   |
| logFile                | 1 = buffer log, 2 = flush after each print<br>type: string   |
| logFileName            | name of log file, if empty, console.log will be used. \$Y = year, \$M = month, \$D = day, \$h = hour, \$m = minute, \$s = second<br>type: string |
| logTimeStamps          | add time stamps to console log<br>type: bool<br>default: 0 (false)   |
| m_helicopterPitch      | helicopter mouse pitch scale<br>type: float  |
| m_helicopterYaw        | helicopter mouse yaw scale<br>type: float  |
| m_pitch                | mouse pitch scale<br>type: float   |



| <b>cvar</b>                     | <b>Description</b>  |
|---------------------------------|---|
| m_showMouseRate                 | shows mouse movement<br>type: bool (cheat)<br>default: 0 (false)                        |
| m_smooth                        | number of samples blended for mouse viewing<br>type: int [1, 8]                         |
| m_strafeScale                   | mouse strafe movement scale<br>type: float  |
| m_strafeSmooth                  | number of samples blended for mouse moving<br>type: int [1, 8]                          |
| m_yaw                           | mouse yaw scale<br>type: float  |
| msv_useSampleGame               | use the sample game instead of the etqw SKU<br>type: bool (cheat)<br>default: 0 (false) |
| net_allowCheats                 | Allow cheats in network game<br>type: bool<br>default: 0 (false)                        |
| net_aorPVSScale                 | AoR scale for outside of PVS<br>type: float (cheat)                                     |
| net_autoLANScan                 | automatically scan for LAN servers<br>type: int [0, 60]                                 |
| net_channelShowDrop             | show dropped packets<br>type: bool (cheat)<br>default: 0 (false)                        |
| net_channelShowPackets          | show all packets<br>type: bool (cheat)<br>default: 0 (false)                            |
| net_checkSync                   | enable data synchronization<br>type: bool (cheat)<br>default: 1 (true)                  |
| net_clientAORFilter             | type: string (cheat)  |
| net_clientLagOMeter             | draw prediction graph<br>type: bool<br>default: 0 (false)                               |
| net_clientMaxPrediction         | maximum number of milliseconds a client can predict ahead of server.<br>type: int       |
| net_clientMaxRate               | maximum rate requested by client from server in bytes/sec<br>type: int                  |
| net_clientPrediction            | additional client side prediction in milliseconds<br>type: int                          |
| net_clientPunkbusterEnabled     | is client side PunkBuster enabled?<br>type: bool<br>default: 0 (false)                  |
| net_clientRemoteConsoleAddress  | remote console address<br>type: string  |
| net_clientRemoteConsolePassword | remote console password<br>type: string   |

| <b>cvar</b>                        | <b>Description</b>   |
|------------------------------------|--|
| net_clientSelfSmoothing            | smooth local client position<br>type: bool (cheat)<br>default: 1 (true)  |
| net_clientServerTimeout            | server time out in seconds<br>type: int  |
| net_clientShowAOR                  | type: int [0, 3] (cheat)   |
| net_clientShowSnapshot             | type: int [0, 4] (cheat)   |
| net_clientShowSnapshotRadius       | type: float (cheat)  |
| net_clientSkipUsercmd              | skip ucmds generation -<br>0: no skipping<br>1: no skipping. when multiple game frames, send single packet<br>2: skip. when multiple game frames send only first<br>3: skip. when multiple game frames send only last<br>type: int |
| net_clientUsercmdBackup            | number of usercmds to resend<br>type: int [0, 10]  |
| net_demoErrorOnInvalidUserCmds     | Will cause an error instead of a warning when playing back a demo with no usercmds<br>type: bool (cheat)<br>default: 0 (false)   |
| net_forceDrop                      | percentage packet loss<br>type: int (cheat)  |
| net_forceLatency                   | milliseconds latency<br>type: int (cheat)  |
| net_httpPassword                   | Password used with HTTP authentication to download server assets<br>type: string (cheat)   |
| net_httpProxy                      | Use an http proxy to download game content from servers. Use <a href="#">http://proxy_address:proxy_port</a> . See net_httpProxyUser, net_httpProxyPassword, net_httpProxyMode<br>type: string                                     |
| net_httpProxyMode                  | Proxy configuration mode, 0 - don't use a proxy, 1 - attempt to read from environment (over-write cvar settings on launch), 2 - use cvar settings<br>type: int   |
| net_httpProxyPassword              | Password used with http proxy authentication<br>type: string (cheat)   |
| net_httpProxyUser                  | Username used with http proxy authentication<br>type: string (cheat)   |
| net_httpUser                       | Username used with HTTP authentication to download server assets<br>type: string (cheat)   |
| net_ip                             | IP address of the local network card to be listened on<br>type: string   |
| net_limitApparentMaxErrorAllowance | fraction of the current physics speed added to the maximum apparent speed due to client prediction error<br>type: float  |
| net_limitApparentMaxLagAllowance   | fraction of the current physics speed added to the maximum apparent speed due to client lag<br>type: float   |

| <b>cvar</b>                      | <b>Description</b>   |
|----------------------------------|--|
| net_limitApparentMinSpeed        | minimum value for maximum apparent speed to reach<br>type: float   |
| net_limitApparentVelocity        | limit the apparent velocity of objects in prediction to realistic levels<br>type: bool<br>default: 1 (true)            |
| net_maxPlayerCollisionMerge      | maximum number of player collision steps to merge together during client reprediction<br>type: int                     |
| net_port                         | local IP port number<br>type: int  |
| net_predictionErrorDecay         | Enable/disable prediction error decay<br>type: bool<br>default: 1 (true)   |
| net_serverAllowServerMod         | allow server-side mods<br>type: bool<br>default: 0 (false)   |
| net_serverBalanceSnapshotTimes   | balance snapshot times for a good repartition over game frames<br>type: bool<br>default: 1 (true)                      |
| net_serverClientTimeout          | client time out in seconds<br>type: int  |
| net_serverDedicated              | 1 = text console dedicated server, 2 = graphical dedicated server<br>type: int [0, 2] (init)                           |
| net_serverDrawClient             | number of client for which to draw view on server<br>type: int (cheat)   |
| net_serverLicenseCodeFile        | If not empty, allows license code to be stored outside of the game file system<br>type: string                         |
| net_serverMaxClientRate          | maximum rate to a client in bytes/sec<br>type: int   |
| net_serverMaxReservedClientSlots | maximum number of player slots reserved for session invites<br>type: int [0, 32] (cheat)                               |
| net_serverMaxUsercmdRelay        | maximum number of usercmds from other clients the server relays to a client<br>type: int [1, 10]                       |
| net_serverPunkbusterEnabled      | is server side PunkBuster enabled?<br>type: bool<br>default: 0 (false)   |
| net_serverReloadEngine           | perform a full reload on next map restart (including flushing referenced pak files) - decreased if &gt; 0<br>type: int |
| net_serverRemoteConsolePassword  | remote console password<br>type: string  |
| net_serverSnapshotDelay          | delay between snapshots in number of game frames<br>type: int  |
| net_serverStoreLicenseCode       | Store the license code for dedicated server on disk (1: encrypted, 2: not encrypted)<br>type: int                      |

| <b>cvar</b>                   | <b>Description</b>  |
|-------------------------------|---|
| net_serverZombieTimeout       | disconnected client timeout in seconds<br>type: int   |
| net_showAsyncStats            | show async network stats<br>type: bool<br>default: 0 (false)  |
| net_socksEnabled              | type: bool<br>default: 0 (false)  |
| net_socksPassword             | type: string  |
| net_socksPort                 | type: int   |
| net_socksServer               | type: string  |
| net_socksUsername             | type: string  |
| net_staggerPlayerGroundChecks | skip every other ground check during forward prediction<br>type: bool<br>default: 1 (true)                                      |
| net_updateAutoDownload        | control auto download of game updates. 0: no, 1: prompt on client, auto download on server 2: always auto download<br>type: int |
| net_updateAutoExecute         | execute the installer once downloaded. 0: no, 1: yes on client, not on server, 2: yes<br>type: int                              |
| net_useAOR                    | Enable/Disable Area of Relevance<br>type: bool (cheat)<br>default: 1 (true)   |
| net_useUPnP                   | Use UPnP for external address identification<br>type: bool<br>default: 1 (true)   |
| net_verbose                   | 1 = verbose output, 2,3 = even more verbose output<br>type: int [0, 3]  |
| password                      | client password used when connecting<br>type: string  |
| pm_bboxwidth                  | x/y size of player's bounding box<br>type: float  |
| pm_bobpitch                   | type: float   |
| pm_bobroll                    | type: float   |
| pm_bobup                      | type: float   |
| pm_crouchbob                  | bob much faster when crouched<br>type: float  |
| pm_crouchheight               | height of player's bounding box while crouched<br>type: float   |
| pm_crouchrage                 | time it takes for player's view to change from standing to crouching<br>type: float   |
| pm_crouchspped                | speed the player can move while crouched<br>type: float   |
| pm_crouchviewheight           | height of player's view while crouched<br>type: float   |
| pm_deadheight                 | height of player's bounding box while dead<br>type: float   |

| <b>cvar</b>                | <b>Description</b>  |
|----------------------------|---|
| pm_deadviewheight          | height of player's view while dead<br>type: float   |
| pm_deathThirdPersonAngle   | direction of camera from player in 3rd person in degrees (0 = behind player, 180 = in front)<br>type: float (cheat)                             |
| pm_deathThirdPersonHeight  | height of camera from normal view height in 3rd person<br>type: float (cheat)   |
| pm_deathThirdPersonRange   | camera distance from player in 3rd person<br>type: float (cheat)  |
| pm_democamspeed            | speed the player can move while flying around in a demo<br>type: float (cheat)  |
| pm_deployThirdPersonAngle  | direction of camera from player in 3rd person in degrees (0 = behind player, 180 = in front)<br>type: float (cheat)                             |
| pm_deployThirdPersonHeight | height of camera from normal view height in 3rd person<br>type: float (cheat)   |
| pm_deployThirdPersonRange  | camera distance from player in 3rd person<br>type: float (cheat)  |
| pm_friction                | friction applied to player on the ground<br>type: float   |
| pm_jumpheight              | approximate height the player can jump<br>type: float   |
| pm_maxproneviewpitch       | amount player's view can look down when prone<br>type: float  |
| pm_maxviewpitch            | amount player's view can look down<br>type: float   |
| pm_minproneviewpitch       | amount player's view can look up when prone(negative values are up)<br>type: float  |
| pm_minviewpitch            | amount player's view can look up (negative values are up)<br>type: float  |
| pm_noclipspeed             | speed the player can move while in noclip<br>type: float  |
| pm_noclipspeedsprint       | speed the player can move while in noclip and sprinting<br>type: float  |
| pm_noclipspeedwalk         | speed the player can move while in noclip and walking<br>type: float  |
| pm_normalheight            | height of player's bounding box while standing<br>type: float   |
| pm_normalviewheight        | height of player's view while standing<br>type: float   |
| pm_pausePhysics            | pauses physics<br>type: bool<br>default: 0 (false)  |
| pm_powerslide              | adjust the push when pm_slidevelocity == 1, set power &lt; 1 - &gt; more speed, &gt; 1 -&gt; closer to pm_slidevelocity 0<br>type: float [0, 4] |
| pm_proneheight             | height of player's bounding box while prone<br>type: float  |

| <b>cvar</b>            | <b>Description</b>  |
|------------------------|---|
| pm_pronespeed          | speed the player can move while prone<br>type: float  |
| pm_proneviewdistance   | distance in front of the player's view while prone<br>type: float   |
| pm_proneviewheight     | height of player's view while prone<br>type: float  |
| pm_runbob              | bob faster when running<br>type: float  |
| pm_runpitch            | type: float   |
| pm_runroll             | type: float   |
| pm_runspeed            | speed the player can move while running<br>type: float  |
| pm_runspeedback        | speed the player can move backwards while running<br>type: float  |
| pm_runspeedforward     | speed the player can move forwards while running<br>type: float   |
| pm_runspeedstrafe      | speed the player can move sideways while running<br>type: float   |
| pm_slidevelocity       | what to do with velocity when hitting a surface at an angle. 0: use horizontal speed, 1: keep some of the impact speed to push along the slide<br>type: bool<br>default: 1 (true) |
| pm_spectatebbox        | size of the spectator bounding box<br>type: float   |
| pm_spectatespeed       | speed the player can move while spectating<br>type: float   |
| pm_spectatespeedsprint | speed the player can move while spectating and sprinting<br>type: float   |
| pm_sprintspeed         | speed the player can move while sprinting<br>type: float  |
| pm_sprintspeedforward  | speed the player can move forwards while sprinting<br>type: float   |
| pm_sprintspeedstrafe   | speed the player can move sideways while sprinting<br>type: float   |
| pm_stepsize            | maximum height the player can step up without jumping<br>type: float  |
| pm_thirdPerson         | enables third person view<br>type: bool (cheat)<br>default: 0 (false)   |
| pm_thirdPersonAngle    | direction of camera from player in 3rd person in degrees (0 = behind player, 180 = in front)<br>type: float (cheat)   |
| pm_thirdPersonClip     | clip third person view into world space<br>type: bool (cheat)<br>default: 1 (true)  |
| pm_thirdPersonHeight   | height of camera from normal view height in 3rd person<br>type: float (cheat)   |

| <b>cvar</b>              | <b>Description</b>   |
|--------------------------|--|
| pm_thirdPersonNoPitch    | ignore camera pitch when in third person mode<br>type: bool (cheat)<br>default: 0 (false)  |
| pm_thirdPersonOrbit      | if set, will automatically increment pm_thirdPersonAngle every frame<br>type: float (cheat)  |
| pm_thirdPersonRange      | camera distance from player in 3rd person<br>type: float (cheat)   |
| pm_vehicleSoundLerpScale | type: float  |
| pm_walkbob               | bob slowly when walking<br>type: float   |
| pm_walkspeed             | speed the player can move while walking<br>type: float   |
| pm_waterFloatValue       | fraction of water coverage at which the player will try to float<br>type: float (cheat)  |
| pm_waterSpeed            | speed player will be pushed up in water when totally under water<br>type: float (cheat)  |
| r_32ByteVtx              | Uses 32bit vtx<br>type: string (cheat)   |
| r_ambientARB             | 0 = ambient cube map rendering done through ff pipeline, 1 = ambient cube map rendering done through ARB program<br>type: bool (cheat)<br>default: 0 (false) |
| r_ambientLightDrawMode   | 0 is blended sphere, 1 is circle outline, 2 is completely off<br>type: int (cheat)   |
| r_ambientMap             | ambient cube map base name<br>type: string (cheat)   |
| r_ambientMapNoUpdate     | disabled dynamic ambient cubemap texture uploading<br>type: string (cheat)   |
| r_ambientScale           | ambient cube mapping brightness<br>type: float (cheat)   |
| r_aspectRatio            | aspect ratio. 0 is 4:3, 1 is 16:9, 2 is 16:10, 3 is 5:4. -1 uses r_customAspectRatioH and r_customAspectRatioV<br>type: int                                  |
| r_atmosScale             | atmosphere brightness<br>type: float (cheat)   |
| r_brightness             | changes gamma tables<br>type: float [0.5, 2]   |
| r_checkBounds            | compare all surface bounds with precalculated ones<br>type: bool (cheat)<br>default: 0 (false)   |
| r_clear                  | force screen clear every frame, 1 = purple, 2 = black, 'r g b' = custom<br>type: string (cheat)  |
| r_combinedReflectDist    | Distance where the planar reflection gets cut of<br>type: string (cheat)   |
| r_customAspectRatioH     | horizontal custom aspect ratio<br>type: float  |

| <b>cvar</b>                 | <b>Description</b>   |
|-----------------------------|--|
| r_customAspectRatioV        | vertical custom aspect ratio<br>type: float  |
| r_customHeight              | custom screen height<br>type: int  |
| r_customWidth               | custom screen width<br>type: int   |
| r_debugArrowStep            | step size of arrow cone line rotation in degrees<br>type: int [0, 120]                               |
| r_debugAxisLength           | used to set the length of drawn debug axis<br>type: string (cheat)                                   |
| r_debugEntityRefs           | type: int (cheat)  |
| r_debugImposters            | Show some extra information on imposter rendering<br>type: bool (cheat)<br>default: 0 (false)        |
| r_debugLineDepthTest        | perform depth test on debug lines<br>type: bool<br>default: 0 (false)                                |
| r_debugLineWidth            | width of debug lines<br>type: bool<br>default: 1 (true)  |
| r_debugPolygonFilled        | draw a filled polygon<br>type: bool (cheat)<br>default: 1 (true)                                     |
| r_debugViewEntities         | type: int (cheat)  |
| r_depthFill                 | Enable depth only pass<br>type: string (cheat)   |
| r_depthFillCutoff           | Screen Rect Area required to render<br>type: string (cheat)  |
| r_depthFillNoColour         | Disable depth fill colour write<br>type: string (cheat)  |
| r_depthRangeStartDefault    | type: string (cheat)   |
| r_depthRangeWeaponHackEnd   | type: string (cheat)   |
| r_depthRangeWeaponHackScale | type: string (cheat)   |
| r_detailFade                | Distance fading control ('reloadImages all' needed)<br>type: float [0, 1]                            |
| r_detailRatio               | Ratio of detail texture to main texture<br>type: int (cheat)   |
| r_detailTexture             | Detail texture on landscape ('reparseDecls renderProgram' needed)<br>type: bool<br>default: 1 (true) |
| r_displayRefresh            | optional display refresh rate option for vid mode<br>type: int [0, 200]                              |
| r_dumpShaders               | Dump compiled and preprocessed shaders to text files<br>type: string (cheat)                         |
| r_dynamicModelCacheMegs     | Number of megabytes to cache dynamic model instantiations in.<br>type: int (cheat)                   |



| <b>cvar</b>               | <b>Description</b>   |
|---------------------------|--|
| r_farPlaneDist            | Optional far plane clipping<br>type: float (cheat)   |
| r_finish                  | force a call to glFinish() every frame<br>type: bool (cheat)<br>default: 0 (false)                         |
| r_flareSize               | scale the flare deforms from the material def<br>type: float (cheat)                                       |
| r_flushPurgeableModels    | Delete all cached purgeable models<br>type: bool (cheat)<br>default: 0 (false)                             |
| r_forceDistancePortalCull | type: float (cheat)  |
| r_forceGLFinish           | force finish within backend<br>type: int (cheat)   |
| r_forceLoadImages         | draw all images to screen after registration<br>type: bool<br>default: 0 (false)                           |
| r_frontBuffer             | draw to front buffer for debugging<br>type: bool (cheat)<br>default: 0 (false)                             |
| r_fullscreen              | 0 = windowed, 1 = full screen<br>type: bool<br>default: 1 (true)   |
| r_gamma                   | changes gamma tables<br>type: float [0.5, 3]   |
| r_generateInteractions    | Generate all interactions after a map load<br>type: string (cheat)   |
| r_glDriver                | „ „opengl32“ „, etc.<br>type: string (cheat)“  |
| r_glDriverVendor          | OpenGL driver vendor<br>type: string (cheat)   |
| r_glowbase                | base glare level<br>type: float (cheat)  |
| r_ignore                  | used for random debugging without defining new vars<br>type: string (cheat)                                |
| r_ignore2                 | used for random debugging without defining new vars<br>type: string (cheat)                                |
| r_ignoreGLErrors          | ignore GL errors<br>type: bool (cheat)<br>default: 1 (true)  |
| r_imposterCoverage        | type: float (cheat)  |
| r_imposterCutoff          | Size in pixels at which imposters stop drawing<br>type: int (cheat)  |
| r_imposterFadeEnd         | Size in pixels at which imposters are fully faded in (and the original stops drawing)<br>type: int (cheat) |
| r_imposterFadeStart       | Size in pixels at which imposters start to fade in<br>type: int (cheat)                                    |
| r_inhibitFBO              | Disable the fbo extension.<br>type: bool (cheat)<br>default: 0 (false)                                     |

| <b>cvar</b>                | <b>Description</b>  |
|----------------------------|---|
| r_jitter                   | randomly subpixel jitter the projection matrix<br>type: bool (cheat)<br>default: 0 (false)              |
| r_jitterBias               | jitter radius bias<br>type: float [0, 1] (cheat)  |
| r_jitterRadial             | radial jitter view matrix around the specified distance<br>type: float (cheat)                          |
| r_jitterRadius             | offset from screen origin for jitter<br>type: float (cheat)   |
| r_jointNameOffset          | offset of joint names when r_showskel is set to 1<br>type: float (cheat)                                |
| r_jointNameScale           | size of joint names when r_showskel is set to 1<br>type: float (cheat)                                  |
| r_keepOldViewEntities      | type: bool (cheat)<br>default: 0 (false)  |
| r_lightAllBackFaces        | light all the back faces, even when they would be shadowed<br>type: bool (cheat)<br>default: 0 (false)  |
| r_lightScale               | all light intensities are multiplied by this<br>type: float (cheat)                                     |
| r_lightSourceRadius        | for soft-shadow sampling<br>type: float (cheat)   |
| r_lispAlternateOrigin      | Use alternative ways of calculating the shadow view origin.<br>type: int (cheat)                        |
| r_lispAlternateZfar        | Use a different calculation for the warp factor.<br>type: int (cheat)                                   |
| r_lispAngle                | Angle to rotate the shadowview in ndc space<br>type: int (cheat)  |
| r_lispCorrectForViewAngle  | Correct warp factor based on the angle with the light.<br>type: int (cheat)                             |
| r_lispDebugFrustum         | Allow the use of the existing projeciton matrix to debug certain lightspace cases.<br>type: int (cheat) |
| r_lispflipx                | Test some stuff<br>type: float (cheat)  |
| r_lispflipy                | Test some stuff<br>type: float (cheat)  |
| r_lispflipz                | Test some stuff<br>type: float (cheat)  |
| r_lisptest                 | Test some stuff<br>type: int (cheat)  |
| r_lispUseBodyVector        | Use body vector instead of the view vector<br>type: bool (cheat)<br>default: 1 (true)                   |
| r_listMissingInstanceProgs | list mia render programs for instancing<br>type: bool (cheat)<br>default: 0 (false)                     |
| r_loadAnimB                | Attempt loading of binary version of animations.<br>type: bool (cheat)<br>default: 1 (true)             |

| <b>cvar</b>              | <b>Description</b>   |
|--------------------------|--|
| r_loadModelBs            | allow loading of modelb render models<br>type: bool (cheat)<br>default: 1 (true)   |
| r_lockDebugLines         | type: bool (cheat)<br>default: 0 (false)   |
| r_lockedView             | type: int (cheat)  |
| r_lockSurfaces           | allow moving the view point without changing the composition of the scene, including culling<br>type: bool (cheat)<br>default: 0 (false) |
| r_logFile                | number of frames to emit GL logs<br>type: int (cheat)  |
| r_mapShotAngles          | Pitch Yaw Roll of mapshot angles<br>type: string (cheat)   |
| r_materialOverride       | overrides all materials<br>type: string (cheat)  |
| r_matey                  | AVAST!<br>type: string (cheat)   |
| r_maxViewEntities        | type: int (cheat)  |
| r_MD5FreeScaledAndBase   | type: bool (cheat)<br>default: 1 (true)  |
| r_MD5LoadBinary          | type: bool (cheat)<br>default: 0 (false)   |
| r_MD5LodAltLodNaming     | type: bool (cheat)<br>default: 1 (true)  |
| r_MD5LodBias             | LOD bias for a MD5 mesh<br>type: int (cheat)   |
| r_MD5LodPrint            | Print a message when an LOD state changes<br>type: bool (cheat)<br>default: 0 (false)  |
| r_MD5LodScale            | LOD scaling for a MD5 mesh<br>type: float  |
| r_MD5MaxLodStages        | maximum number of LOD stages loaded for a MD5 mesh<br>type: int (cheat)  |
| r_MD5UseDrawVertFast     | type: bool (cheat)<br>default: 1 (true)  |
| r_MD5UseHardwareSkinning | Do MD5 skinning partially on the GPU<br>type: bool (read only)<br>default: 1 (true)  |
| r_MD5WriteBinary         | type: bool (cheat)<br>default: 0 (false)   |
| r_megaDrawMethod         | type: int  |
| r_megaFadeTime           | type: int  |
| r_megaShowGrid           | show a grid pattern<br>type: bool (cheat)<br>default: 0 (false)  |
| r_megaShowTileSize       | show tile sizes<br>type: bool (cheat)<br>default: 0 (false)  |
| r_megaStreamBlocks       | type: int [1, 32]  |

| <b>cvar</b>            | <b>Description</b>  |
|------------------------|---|
| r_megaStreamFromDVD    | type: bool<br>default: 0 (false)  |
| r_megaTilesPerSecond   | only update a limited number of mega texture tiles per second<br>type: int  |
| r_megaUpscale          | type: bool<br>default: 0 (false)  |
| r_mergeModelSurfaces   | combine model surfaces with the same material<br>type: bool (cheat)<br>default: 1 (true)                              |
| r_mirrorplaneslop      | Amount to offset reflection clipping planes<br>type: string (cheat)   |
| r_mode                 | video mode number<br>type: int  |
| r_multiSamples         | number of antialiasing samples<br>type: int   |
| r_noDoubleAtmosphere   | Uses the stencil buffer to avoid atmosphere-ing<br>type: string (cheat)   |
| r_normalizeNormalMaps  | Normalize normalmaps after lookup.<br>type: bool<br>default: 1 (true)   |
| r_notchDir             | refraction plane offset direction<br>type: float (cheat)  |
| r_occlusionBBDebug     | Display occlusion BB test<br>type: int (cheat)  |
| r_occlusionCutoff      | Don't do occlusion queries if screen space in pixels is bigger than this<br>type: int (cheat)                         |
| r_occlusionDebug       | Print debug info on occlusion queries, 2 = skip all oq gl commands, 3 = skip oq readback command<br>type: int (cheat) |
| r_occlusionFlush       | Use explicit glFlush after firing off queries<br>type: bool (cheat)<br>default: 0 (false)                             |
| r_occlusionSystem      | 0 = Clever, 1 = Frame delay<br>type: int (cheat)  |
| r_occlusionThreshold   | Consider object as occluded if less or equal than x pixels are visible.<br>type: int (cheat)                          |
| r_occlusionWaitPredict | Predict as objects as visibe when waiting for result.<br>type: bool (cheat)<br>default: 1 (true)                      |
| r_offsetfactor         | polygon offset parameter<br>type: float (cheat)   |
| r_offsetunits          | polygon offset parameter<br>type: float (cheat)   |
| r_orderIndexes         | perform index reorganization to optimize vertex use<br>type: bool (cheat)<br>default: 1 (true)                        |
| r_refractFovChange     | FOV modification used during refraction subview rendering<br>type: float (cheat)                                      |

| <b>cvar</b>                | <b>Description</b>   |
|----------------------------|--|
| r_refractOffset            | refraction plane offset<br>type: float (cheat)   |
| r_renderProgramLodDistance | Sets global render program lod distance<br>type: float   |
| r_renderProgramLodFade     | Sets global render program fade distance<br>type: float (cheat)  |
| r_screenFraction           | for testing fill rate, the resolution of the entire screen can be changed<br>type: int (cheat)   |
| r_selectedAmbientLight     | Which ambient light should be drawn as selected<br>type: int (cheat)   |
| r_shaderPreferALU          | Use ALU instructions instead of textures in shaders.<br>type: bool<br>default: 1 (true)  |
| r_shaderQuality            | Sets the level of detail to use for shaders, 0 = highest<br>type: int  |
| r_shaderSkipSpecCubeMaps   | Use specular cube maps.<br>type: bool<br>default: 0 (false)  |
| r_shadowPass               | enable shadow pass<br>type: bool (cheat)<br>default: 1 (true)  |
| r_shadowPolygonFactor      | scale value for stencil shadow drawing<br>type: float  |
| r_shadowPolygonOffset      | bias value added to depth test for stencil shadow drawing<br>type: float   |
| r_shadows                  | enable shadows<br>type: bool<br>default: 1 (true)  |
| r_showAlloc                | report alloc/free counts<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showAmbientLights        | show ambient light sources<br>type: string (cheat)   |
| r_showBackendSpeeds        | show the backend timing split<br>type: int (cheat)   |
| r_showBatches              | draw each batch with a different color<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showBatchInfo            | report batch counts for various types<br>type: int (cheat)   |
| r_showBatchSize            | highlighting of draw batches that fall below the desired polygons per draw call (purple really bad, red bad, yellow moderate, green good), 1 = only draw visible ones, 2 = draw all front facing, 3 = draw all<br>type: int [0, 3] (cheat) |
| r_showCull                 | report sphere and box culling stats<br>type: bool (cheat)<br>default: 0 (false)  |
| r_showCullSectors          | Show cull sectors<br>type: int (cheat)   |

| <b>cvar</b>               | <b>Description</b>   |
|---------------------------|--|
| r_showDefaultRenderEnts   | type: string (cheat)   |
| r_showDefs                | report the number of modeDefs and lightDefs in view<br>type: bool (cheat)<br>default: 0 (false)  |
| r_showDemo                | report reads and writes to the demo file<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showDepth               | display the contents of the depth buffer and the depth range<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showDominantTri         | draw lines from vertexes to center of dominant triangles<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showDrawOrder           | type: int (cheat)  |
| r_showDynamic             | report stats on dynamic surface generation<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showDynamicDefs         | Show dynamic updating entities i.e. freeing cached models and/or shadow volumes ( purple = entity def changed, cyan = model callback forced update )<br>type: string (cheat) |
| r_showEdges               | draw the sil edges<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showEntityScissors      | show entity scissor rectangles<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showEnvBounds           | Show env bounds<br>type: int (cheat)   |
| r_showFacePlanes          | shows shadow volume face planes<br>type: float (cheat)   |
| r_showGUISpeeds           | report time spent drawing GUIs<br>type: int (cheat)  |
| r_showImages              | 1 = show all images instead of rendering, 2 = show in proportional size<br>type: int [0, 2] (cheat)  |
| r_showIndexTreeBounds     | Show index tree bounds<br>type: int (cheat)  |
| r_showInstanceInfo        | Show Instance Info<br>type: int (cheat)  |
| r_showIntensity           | draw the screen colors based on intensity, red = 0, green = 128, blue = 255<br>type: bool (cheat)<br>default: 0 (false)  |
| r_showInteractionFrustums | 1 = show a frustum for each interaction, 2 = also draw lines to light origin, 3 = also draw entity bbox<br>type: int [0, 3] (cheat)  |
| r_showInteractions        | report interaction generation activity<br>type: bool (cheat)<br>default: 0 (false)   |

| <b>cvar</b>               | <b>Description</b>   |
|---------------------------|--|
| r_showInteractionScissors | 1 = show screen rectangle which contains the interaction frustum, 2 = also draw construction lines<br>type: int [0, 2] (cheat)   |
| r_showLightCount          | 1 = colors surfaces based on light count, 2 = also count everything through walls, 3 = also print overdraw<br>type: int [0, 3] (cheat)                                   |
| r_showLights              | 1 = just print volumes numbers, highlighting ones covering the view, 2 = also draw planes of each volume, 3 = also draw edges of each volume<br>type: int [0, 3] (cheat) |
| r_showLightScale          | report the scale factor applied to drawing for overbrights<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showLightScissors       | show light scissor rectangles<br>type: bool (cheat)<br>default: 0 (false)  |
| r_showMegaTexture         | report megatexture upload/bind counts<br>type: bool (cheat)<br>default: 0 (false)  |
| r_showMegaTextureLevels   | Display the different levels using color coded textures.<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showMegaTextureSTGrid   | Show megaTexture ST grid<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showMemory              | print frame memory utilization<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showModelTypes          | show model types, red - md5, blue - area, green - lodentity, white - other<br>type: int (cheat)  |
| r_showNodePlanes          | enables portal plane drawing<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showNormals             | draws wireframe normals<br>type: float (cheat)   |
| r_showOcclusions          | Show occlusion queries culled by occlusion queries<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showOverDraw            | 1 = geometry overdraw, 2 = light interaction overdraw, 3 = geometry and light interaction overdraw<br>type: int [0, 3] (cheat)   |
| r_showParentFrustum       | Shows the frustum of the parent view in subview and shadow renders.<br>type: bool (cheat)<br>default: 0 (false)  |
| r_showPortals             | draw portal outlines in color based on passed / not passed<br>type: int (cheat)  |
| r_showPrimitives          | report drawsurf/index/vertex counts<br>type: int (cheat)   |

| <b>cvar</b>            | <b>Description</b>  |
|------------------------|---|
| r_showQuadTrees        | draw quadtree outlines in color, 1 = 3D, 2 = 2D, 3 = depth buffered 3D<br>type: int (cheat)   |
| r_showQueryTimers      | Show the query_timer extension results.<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showShadowCount      | colors screen based on shadow volume depth complexity, &gt;= 2 = print overdraw count based on stencil index values, 3 = only show turboshadows, 4 = only show static shadows<br>type: int [0, 4] (cheat) |
| r_showShadows          | 1 = visualize the stencil shadow volumes, 2 = draw filled in<br>type: int [0, 4] (cheat)  |
| r_showSilhouette       | highlight edges that are casting shadow planes<br>type: bool (cheat)<br>default: 0 (false)  |
| r_showSkel             | draw the skeleton when model animates, 1 = draw model with skeleton, 2 = draw skeleton only<br>type: int [0, 2] (cheat)   |
| r_showSmp              | show which end (front or back) is blocking<br>type: bool (cheat)<br>default: 0 (false)  |
| r_showStuffCache       | Show allocation events of the stuff cache<br>type: bool (cheat)<br>default: 0 (false)   |
| r_showSurfaceAllocated | show surface allocated 1 - planes, 2 - silIndexes, 3 - silEdges, 4 - shadowVertexes, 5 - weightCache<br>type: int (cheat)   |
| r_showSurfaceInfo      | show surface material name under crosshair<br>type: int (cheat)   |
| r_showSurfaces         | report surface/light/shadow counts<br>type: bool (cheat)<br>default: 0 (false)  |
| r_showTangentSpace     | shade triangles by tangent space, 1 = use 1st tangent vector, 2 = use 2nd tangent vector, 3 = use normal vector<br>type: int [0, 3] (cheat)   |
| r_showTexelDensity     | draws all triangles with texel density as gradient<br>type: bool (cheat)<br>default: 0 (false)  |
| r_showTextureMemory    | Show texture memory usage: 1 = total; 2 = per frame<br>type: int (cheat)  |
| r_showTexturePolarity  | shade triangles by texture area polarity<br>type: bool (cheat)<br>default: 0 (false)  |
| r_showTextureVectors   | if &gt; 0 draw each triangles texture (tangent) vectors<br>type: float (cheat)  |
| r_showTrace            | show the intersection of an eye trace with the world<br>type: int (cheat)   |
| r_showTris             | enables wireframe rendering of the world, 1 = only draw visible ones, 2 = draw all front facing, 3 = draw all<br>type: int [0, 3] (cheat)   |



| <b>cvar</b>              | <b>Description</b>  |
|--------------------------|---|
| r_showUnsmoothedTangents | if 1, put all nvidia register combiner programming in display lists<br>type: bool (cheat)<br>default: 0 (false) |
| r_showUpdates            | report entity and light updates and ref counts<br>type: bool (cheat)<br>default: 0 (false)                      |
| r_showVertexCache        | type: int (cheat)   |
| r_showVertexColor        | draws all triangles with the solid vertex color<br>type: bool (cheat)<br>default: 0 (false)                     |
| r_showviewEntities       | 1 = displays the bounding boxes of all view models, 2 = print index numbers<br>type: int (cheat)                |
| r_singleArea             | only draw the portal area the view is actually in<br>type: bool (cheat)<br>default: 0 (false)                   |
| r_singleEntity           | suppress all but one entity<br>type: int (cheat)  |
| r_singleLight            | suppress all but one light<br>type: int (cheat)   |
| r_singleSurface          | suppress all but one surface on each entity<br>type: int (cheat)  |
| r_singleTriangle         | only draw a single triangle per primitive<br>type: int (cheat)  |
| r_skipAmbient            | bypasses all non-interaction drawing<br>type: bool (cheat)<br>default: 0 (false)                                |
| r_skipAreaSurfaces       | skip rendering of mega texture surfaces<br>type: bool (cheat)<br>default: 0 (false)                             |
| r_skipAtmosInteractions  | skip all light/surface interaction drawing<br>type: int   |
| r_skipAtmosphere         | skips atmosphere pass<br>type: bool (cheat)<br>default: 0 (false)   |
| r_skipBackEnd            | don't draw anything<br>type: bool (cheat)<br>default: 0 (false)   |
| r_skipBlendLights        | skip all blend lights<br>type: bool (cheat)<br>default: 0 (false)   |
| r_skipBump               | uses a flat surface instead of the bump map<br>type: bool (cheat)<br>default: 0 (false)                         |
| r_skipCopyTexture        | do all rendering, but don't actually copyTexSubImage2D<br>type: bool (cheat)<br>default: 0 (false)              |
| r_skipDeforms            | leave all deform materials in their original state<br>type: bool (cheat)<br>default: 0 (false)                  |

| <b>cvar</b>                  | <b>Description</b>  |
|------------------------------|---|
| r_skipDepthAmbient           | uses legacy black depth fill (instead of ambient fill)<br>type: bool<br>default: 0 (false)                              |
| r_skipDiffuse                | 1 = use black for diffuse, 2 = use white for diffuse<br>type: int (cheat)   |
| r_skipDynamicTextures        | don't dynamically create textures<br>type: bool (cheat)<br>default: 0 (false)   |
| r_skipFogLights              | skip all fog lights<br>type: bool (cheat)<br>default: 0 (false)   |
| r_skipFrontEnd               | bypasses all front end work, but 2D gui rendering still draws<br>type: bool (cheat)<br>default: 0 (false)               |
| r_skipImposters              | Skip imposter generation and rendering<br>type: bool (cheat)<br>default: 0 (false)                                      |
| r_skipInstances              | skip rendering of instances<br>type: bool (cheat)<br>default: 0 (false)   |
| r_skipInteractions           | skip all light/surface interaction drawing<br>type: int (cheat)   |
| r_skipLodEntitySurfaces      | skip rendering of mega texture surfaces<br>type: bool (cheat)<br>default: 0 (false)                                     |
| r_skipLWOSurfaces            | skip rendering of mega texture surfaces<br>type: bool (cheat)<br>default: 0 (false)                                     |
| r_skipMD5Surfaces            | skip rendering of mega texture surfaces<br>type: bool (cheat)<br>default: 0 (false)                                     |
| r_skipMegaTexture            | Stop updating the megatexture origin<br>type: bool (cheat)<br>default: 0 (false)  |
| r_skipMegaTextureSurfaces    | skip rendering of mega texture surfaces<br>type: bool (cheat)<br>default: 0 (false)                                     |
| r_skipMegaTextureUpload      | Disable the actual texture uploads (backend loading etc. will still happen)<br>type: bool (cheat)<br>default: 0 (false) |
| r_skipNotMegaTextureSurfaces | skip rendering of non mega texture surfaces<br>type: bool (cheat)<br>default: 0 (false)                                 |
| r_skipOverlays               | skip overlay surfaces<br>type: bool (cheat)<br>default: 0 (false)   |
| r_skipParticles              | 1 = skip all particle systems<br>type: int [0, 1] (cheat)   |

| <b>cvar</b>              | <b>Description</b>   |
|--------------------------|--|
| r_skipRefractCopy        | uses copy of frame buffer<br>type: bool<br>default: 0 (false)  |
| r_skipRender             | skip 3D rendering, but pass 2D<br>type: bool (cheat)<br>default: 0 (false)   |
| r_skipRenderContext      | NULL the rendering context during backend 3D rendering<br>type: bool (cheat)<br>default: 0 (false)                 |
| r_skipROQ                | skip ROQ decoding<br>type: bool (cheat)<br>default: 0 (false)  |
| r_skipShadowviews        | Skips shadow map rendering.<br>type: bool (cheat)<br>default: 0 (false)  |
| r_skipShadowViewsBackend | Skip the rendering but all other setup is done.<br>type: bool (cheat)<br>default: 0 (false)                        |
| r_skipSpecular           | use black for specular<br>type: bool (cheat)<br>default: 0 (false)   |
| r_skipStuff              | Don't draw stuff models<br>type: bool<br>default: 0 (false)  |
| r_skipSubviews           | 1 = don't render any gui elements on surfaces<br>type: int (cheat)   |
| r_skipSuppress           | ignore the per-view suppressions<br>type: bool (cheat)<br>default: 0 (false)                                       |
| r_skipTranslucent        | skip the translucent interaction rendering<br>type: bool (cheat)<br>default: 0 (false)                             |
| r_skipUpdates            | 1 = don't accept any entity or light updates, making everything static<br>type: bool (cheat)<br>default: 0 (false) |
| r_slopNormal             | merge normals that dot less than this<br>type: string (cheat)  |
| r_slopTexCoord           | merge texture coordinates this far apart<br>type: string (cheat)   |
| r_slopVertex             | merge xyz coordinates this far apart<br>type: string (cheat)   |
| r_SMPSharedCtx           | SMP renderer uses shared context<br>type: string (cheat)   |
| r_softParticles          | Enable soft particles<br>type: bool<br>default: 0 (false)  |
| r_sortEffects            | Depth sort effects by their origin<br>type: bool (cheat)<br>default: 1 (true)                                      |

| <b>cvar</b>               | <b>Description</b>   |
|---------------------------|--|
| r_sortInteractions        | Sort interactions per material and VBO.<br>type: bool (cheat)<br>default: 1 (true)                           |
| r_sortParticles           | Enable depth sorting of certain particle systems<br>type: bool (cheat)<br>default: 1 (true)                  |
| r_sortStuff               | Depth sort enable/disable<br>type: bool (cheat)<br>default: 1 (true)   |
| r_stateCache              | check state before upload to drive<br>type: string (cheat)   |
| r_stuffCacheMegs          | Number of megabytes to cache stuff models.<br>type: int (cheat)  |
| r_stuffFadeEnd            | Max vis distance for the stuff models<br>type: float (cheat)   |
| r_stuffFadeStart          | Distance at which stuff starts fading<br>type: float (cheat)   |
| r_stuffLod                | Where lod models stop drawing (they start at r_stuffFadeEnd)<br>type: float (cheat)                          |
| r_stuffUpdateAngle        | Camera needs to rotate more than X degrees for stuff models to update<br>type: string (cheat)                |
| r_stuffUpdateDistance     | Camera needs to move more than X units for stuff models to update<br>type: string (cheat)                    |
| r_subviewOnly             | 1 = don't render main view, allowing subviews to be debugged<br>type: bool (cheat)<br>default: 0 (false)     |
| r_swapInterval            | changes wglSwapInterval<br>type: int   |
| r_testGamma               | if > 0 draw a grid pattern to test gamma levels<br>type: float [0, 195] (cheat)                              |
| r_testGammaBias           | if > 0 draw a grid pattern to test gamma levels<br>type: float (cheat)                                       |
| r_testStepGamma           | if > 0 draw a grid pattern to test gamma levels<br>type: float (cheat)                                       |
| r_trisColor               | sets color of wireframe rendering in RGBA float format or 0xRRGGBBAA hex format<br>type: string              |
| r_useAlphaToCoverage      | Use alpha to coverage.<br>type: bool<br>default: 1 (true)  |
| r_useCachedDynamicModels  | cache snapshots of dynamic models<br>type: bool (cheat)<br>default: 1 (true)                                 |
| r_useClippedLightScissors | 0 = full screen when near clipped, 1 = exact when near clipped, 2 = exact always<br>type: int [0, 2] (cheat) |

| <b>cvar</b>               | <b>Description</b>   |
|---------------------------|--|
| r_useCombinerDisplayLists | put all nvidia register combiner programming in display lists<br>type: bool (cheat)<br>default: 1 (true)                         |
| r_useConstantMaterials    | use pre-calculated material registers if possible<br>type: bool (cheat)<br>default: 1 (true)                                     |
| r_useCulling              | 0 = none, 1 = sphere, 2 = sphere + box<br>type: int [0, 2] (cheat)   |
| r_useDeferredTangents     | defer tangents calculations after deform<br>type: bool (cheat)<br>default: 1 (true)  |
| r_useDepthBoundsTest      | use depth bounds test to reduce shadow fill<br>type: bool (cheat)<br>default: 1 (true)   |
| r_useDestinationScissor   | 1 = use scissor rectangle to only render required fragments during post-processing<br>type: bool (cheat)<br>default: 1 (true)    |
| r_useDitherMask           | Dither out fading geometry<br>type: bool<br>default: 1 (true)  |
| r_useEntityCallbacks      | if 0, issue the callback immediately at update time, rather than defering<br>type: bool (cheat)<br>default: 1 (true)             |
| r_useEntityCulling        | 0 = none, 1 = box<br>type: bool (cheat)<br>default: 1 (true)   |
| r_useEntityScissors       | 1 = use custom scissor rectangle for each entity<br>type: bool (cheat)<br>default: 0 (false)                                     |
| r_useExternalShadows      | 1 = skip drawing caps when outside the light volume, 2 = force to no caps for testing<br>type: int [0, 2] (cheat)                |
| r_useFrustumFarDistance   | if != 0 force the view frustum far distance to this distance<br>type: float (cheat)  |
| r_useIndexBuffers         | use ARB_vertex_buffer_object for indexes<br>type: int [0, 1]   |
| r_useIndexHier            | type: string (cheat)   |
| r_useInteractionCulling   | 1 = cull interactions<br>type: bool (cheat)<br>default: 1 (true)   |
| r_useInteractionScissors  | 1 = use a custom scissor rectangle for each shadow interaction, 2 = also crop using portal scissors<br>type: int [-2, 2] (cheat) |
| r_useInteractionTable     | create a full entityDefs * lightDefs table to make finding interactions faster<br>type: bool (cheat)<br>default: 1 (true)        |

| <b>cvar</b>                      | <b>Description</b>   |
|----------------------------------|--|
| r_useLightCulling                | 0 = none, 1 = box, 2 = exact clip of polyhedron faces, 3 = also areas<br>type: int [0, 3] (cheat)                    |
| r_useLightPortalFlow             | use a more precise area reference determination<br>type: bool (cheat)<br>default: 1 (true)                           |
| r_useLightScissors               | 1 = use custom scissor rectangle for each light<br>type: bool (cheat)<br>default: 1 (true)                           |
| r_useMaxVisDist                  | use the masVisDist entity parameter (on is faster)<br>type: int (cheat)  |
| r_useMegaTextureImageCompression | DXT compress megatexture data before uploading to hardware<br>type: bool (cheat)<br>default: 1 (true)                |
| r_useNodeCommonChildren          | stop pushing reference bounds early when possible<br>type: bool (cheat)<br>default: 1 (true)                         |
| r_useNV20MonoLights              | use pass optimization for mono lights<br>type: int (cheat)   |
| r_useOcclusionQueries            | Use hardware occlusion queries<br>type: bool (cheat)<br>default: 1 (true)  |
| r_useOffscreenScreenshotBuffer   | 1 = use offscreen buffer for screenshots to support post-processing<br>type: bool (cheat)<br>default: 1 (true)       |
| r_useOptimizedShadows            | use the dmap generated static shadow volumes<br>type: bool (cheat)<br>default: 1 (true)                              |
| r_usePortals                     | 1 = use portals to perform area culling, otherwise draw everything<br>type: bool (cheat)<br>default: 1 (true)        |
| r_usePreciseTriangleInteractions | 1 = do winding clipping to determine if each ambiguous tri should be lit<br>type: bool (cheat)<br>default: 0 (false) |
| r_useQuadTree                    | Use a quad tree for the cluster models<br>type: bool (cheat)<br>default: 1 (true)                                    |
| r_useRewriteVbo                  | type: string (cheat)   |
| r_useSampleCoverage              | Use multisample coverage to fade entities.<br>type: bool (cheat)<br>default: 1 (true)                                |
| r_useScissor                     | scissor clip as portals and lights are processed<br>type: bool (cheat)<br>default: 1 (true)                          |
| r_useShadowCulling               | try to cull shadows from partially visible lights<br>type: bool (cheat)<br>default: 1 (true)                         |

| <b>cvar</b>               | <b>Description</b>  |
|---------------------------|---|
| r_useShadowDitherMask     | Dither out fading shadows<br>type: string (cheat)   |
| r_useShadowFastParallel   | use optimized shadow rendering for parallel light sources (doesn't make a speed difference and gives render errors)<br>type: bool (cheat)<br>default: 0 (false) |
| r_useShadowMaps           | enable shadow mapped shadows (will fall back to stencil shadows, reloadDecls + map restart needed)<br>type: bool (cheat)<br>default: 0 (false)                  |
| r_useShadowProjectedCull  | discard triangles outside light volume before shadowing<br>type: bool (cheat)<br>default: 1 (true)  |
| r_useShadowSurfaceScissor | scissor shadows by the scissor rect of the interaction surfaces<br>type: bool (cheat)<br>default: 1 (true)  |
| r_useShadowVisDistMult    | type: float   |
| r_useSilRemap             | consider verts with the same XYZ, but different ST the same for shadows<br>type: bool (cheat)<br>default: 1 (true)  |
| r_useSMP                  | Enabled smp renderer<br>type: string (cheat)  |
| r_useStateCaching         | avoid redundant state changes in GL_*() calls<br>type: bool (cheat)<br>default: 1 (true)  |
| r_useTripleTextureARB     | cards with 3+ texture units do a two pass instead of three pass<br>type: bool (cheat)<br>default: 1 (true)  |
| r_useTwoSidedStencil      | do stencil shadows in one pass with different ops on each side<br>type: bool (cheat)<br>default: 1 (true)   |
| r_useVertexBuffer         | use ARB_vertex_buffer_object for vertexes<br>type: int [0, 1] (cheat)   |
| r_useVertexBufferStream   | Use stream instead of dynamic vbo's ( 0 = static 1 = dynamic 2 = stream)<br>type: int (cheat)   |
| r_useVisDebugARBPrograms  | use vertex/fragment `q` programs to visualize debugging tools<br>type: bool (cheat)<br>default: 1 (true)  |
| r_vertexBufferMegs        | type: int (cheat)   |
| r_visDistMult             | type: float [0.8, 1]  |
| r_visDistOfs              | type: float (cheat)   |
| r_writeAnimB              | Write out binary versions of animations.<br>type: bool (cheat)<br>default: 0 (false)  |
| r_writeModelB             | write out modelb versions of static models<br>type: bool (cheat)<br>default: 0 (false)  |
| r_znear                   | near Z clip plane distance<br>type: float [0.001, 200] (cheat)  |

| <b>cvar</b>          | <b>Description</b>  |
|----------------------|---|
| rb_showActive        | show rigid bodies that are not at rest<br>type: bool (cheat)<br>default: 0 (false)      |
| rb_showBodies        | show rigid bodies<br>type: bool (cheat)<br>default: 0 (false)                           |
| rb_showContacts      | show contact points on rigid bodies<br>type: bool (cheat)<br>default: 0 (false)         |
| rb_showInertia       | show the inertia tensor of each rigid body<br>type: bool (cheat)<br>default: 0 (false)  |
| rb_showMass          | show the mass of each rigid body<br>type: bool (cheat)<br>default: 0 (false)            |
| rb_showTimings       | show rigid body cpu usage<br>type: int (cheat)  |
| rb_showVelocity      | show the velocity of each rigid body<br>type: bool (cheat)<br>default: 0 (false)        |
| s_clipVolumes        | type: bool (cheat)<br>default: 1 (true)   |
| s_constantAmplitude  | type: float (cheat)   |
| s_debugSoundEmitter  | type: int (cheat)   |
| s_decompressionLimit | specifies maximum uncompressed sample length in seconds<br>type: int                    |
| s_doorDistanceAdd    | reduce sound volume with this distance when going through a door<br>type: float (cheat) |
| s_dotbias2           | type: float (cheat)   |
| s_dotbias6           | type: float (cheat)   |
| s_drawSounds         | type: int [0, 4] (cheat)  |
| s_drawSoundScale     | type: float (cheat)   |
| s_driver             | sound driver<br>type: string [best, openal, dsound]                                     |
| s_force22kHz         | type: bool (cheat)<br>default: 0 (false)  |
| s_globalFraction     | volume to all speakers when not spatialized<br>type: float                              |
| s_ignoreALErrors     | ignore OpenAL errors<br>type: bool (cheat)<br>default: 1 (true)                         |
| s_libOpenAL          | OpenAL DLL name/path<br>type: string  |
| s_loadSoundsOnDemand | enable on demand loading of sounds<br>type: bool (init)<br>default: 0 (false)           |



| <b>cvar</b>            | <b>Description</b>  |
|------------------------|---|
| s_maxLowPrioritySounds | Maximum number of low priority sounds which can be active at once<br>type: int                  |
| s_maxSoundsPerShader   | type: int [0, 10]   |
| s_meterTopTime         | type: int   |
| s_minVolume2           | type: float (cheat)   |
| s_minVolume6           | type: float (cheat)   |
| s_noSound              | type: bool (cheat)<br>default: 0 (false)  |
| s_numberOfSpeakers     | number of speakers<br>type: string  |
| s_playDefaultSound     | play a beep for missing sounds<br>type: bool<br>default: 1 (true)                               |
| s_quadraticFalloff     | type: bool (cheat)<br>default: 1 (true)   |
| s_realTimeDecoding     | type: bool (init)<br>default: 1 (true)  |
| s_reverse              | type: bool<br>default: 0 (false)  |
| s_showForcedStopSound  | warn when a sound channel gets forcefully stopped<br>type: bool (cheat)<br>default: 0 (false)   |
| s_showLevelMeter       | type: bool (cheat)<br>default: 0 (false)  |
| s_showPurgedSounds     | print a warning if a purged sample is played<br>type: bool (cheat)<br>default: 0 (false)        |
| s_showStartSound       | type: bool (cheat)<br>default: 0 (false)  |
| s_singleEmitter        | mute all sounds but this emitter<br>type: int (cheat)   |
| s_spatializationDecay  | type: float   |
| s_subFraction          | volume to subwoofer in 5.1<br>type: float   |
| s_useAdpcmCompression  | Use adpcm compression on single channel uncompressed samples<br>type: bool<br>default: 1 (true) |
| s_useDeferredSettings  | use OpenAL deferred settings<br>type: bool (cheat)<br>default: 1 (true)                         |
| s_useOcclusion         | type: bool (cheat)<br>default: 1 (true)   |
| s_voipVolumeIn_dB      | inbound volume adjust for voip in dB<br>type: float   |
| s_voipVolumeOut_dB     | outbound volume adjust for voip volume in dB<br>type: float                                     |
| s_volume_dB            | volume in dB<br>type: float   |

| <b>cvar</b>       | <b>Description</b>   |
|-------------------|--|
| sensitivity       | mouse view sensitivity<br>type: float  |
| si_adminname      | admin name(s)<br>type: string  |
| si_adminStart     | admin required to start the match<br>type: string                                  |
| si_campaign       | current active campaign<br>type: string (read only)                                |
| si_campaignInfo   | current campaign map info<br>type: string (read only)                              |
| si_disableVoting  | disable/enable all voting<br>type: string  |
| si_email          | contact email address<br>type: string  |
| si_irc            | IRC channel<br>type: string  |
| si_map            | current active map<br>type: string (read only)                                     |
| si_maxPlayers     | max number of players allowed on the server<br>type: int [1, 32]                   |
| si_minPlayers     | minimum players before a game can be started<br>type: int                          |
| si_motd_1         | motd line 1<br>type: string  |
| si_motd_2         | motd line 2<br>type: string  |
| si_motd_3         | motd line 3<br>type: string  |
| si_motd_4         | motd line 4<br>type: string  |
| si_name           | name of the server<br>type: string   |
| si_needPass       | enable client password checking<br>type: bool<br>default: 0 (false)                |
| si_privateClients | max number of private players allowed on the server<br>type: int [0, 32]           |
| si_pure           | server is pure and does not allow modified data<br>type: bool<br>default: 1 (true) |
| si_readyPercent   | percentage of players that need to ready up to start a match<br>type: float        |
| si_rules          | ruleset for game<br>type: string   |
| si_spectators     | allow spectators or require all clients to play<br>type: bool<br>default: 1 (true) |

| <b>cvar</b>          | <b>Description</b>  |
|----------------------|---|
| si_teamDamage        | enable team damage<br>type: bool<br>default: 1 (true)   |
| si_teamForceBalance  | Stop players from unbalancing teams<br>type: bool<br>default: 1 (true)  |
| si_timelimit         | time limit (mins)<br>type: float  |
| si_version           | engine version<br>type: string (read only)  |
| si_website           | website info<br>type: string  |
| sm_backOffsetFactor  | Offset factor for shadow buffer rendering.<br>type: float (cheat)   |
| sm_backOffsetUnits   | Offset units for shadow buffer rendering.<br>type: float (cheat)  |
| sm_boundsExpand      | Nudge bounds a bit to account for inaccuracies<br>type: float (cheat)   |
| sm_clipfrustum       | Clip the shadow map frustum to the scene bounds<br>type: bool (cheat)<br>default: 0 (false)                                     |
| sm_farplane          | Frustum far plane for shadow map rendering<br>type: float (cheat)   |
| sm_frontFaces        | Render front faces as well as back faces.<br>type: bool (cheat)<br>default: 0 (false)   |
| sm_frontOffsetFactor | Offset factor for shadow buffer rendering.<br>type: float (cheat)   |
| sm_frontOffsetUnits  | Offset units for shadow buffer rendering.<br>type: float (cheat)  |
| sm_mapping           | Which mapping to use when generating shadow maps.<br>(0=uniform, 1=perspective, 2=light space perspective)<br>type: int (cheat) |
| sm_mediumplane       | Frustum far plane for high res shadow map rendering<br>type: float (cheat)  |
| sm_occlusionQueries  | Use occlusion queries in shadow map rendering.<br>type: bool (cheat)<br>default: 1 (true)                                       |
| sm_renderBackFaces   | Use backfaces to render shadow maps<br>type: bool (cheat)<br>default: 1 (true)  |
| sys_arch             | type: string (init)   |
| sys_cpuHTStatus      | type: string (init)   |
| sys_cpuLogicalCount  | type: int (init)  |
| sys_cpuPhysicalCount | type: int (init)  |
| sys_cpuSpeed         | type: int (init)  |
| sys_cpuString        | type: string (init)   |
| sys_lang             | type: string [english, spanish, italian, german, french, russian, polish, korean, japanese, chinese]                            |
| sys_ram              | type: int (init)  |

| <b>cvar</b>                | <b>Description</b>   |
|----------------------------|--|
| sys_videoRam               | type: int (init)   |
| testLightColor             | the light color to be used for a 'testlight'<br>type: string   |
| timescale                  | scales the time<br>type: float [0.1, 10]   |
| ui_advancedDrivingControls | if true, advanced driving controls are activated<br>type: bool<br>default: 0 (false)                                       |
| ui_advancedFlightControls  | if true, advanced flight controls are activated<br>type: bool<br>default: 0 (false)  |
| ui_autoSwitchEmptyWeapons  | if true, will switch to the next usable weapon when the current weapon runs out of ammo<br>type: bool<br>default: 1 (true) |
| ui_clanTag                 | player clan tag<br>type: string  |
| ui_clanTagPosition         | positioning of player clan tag. 0 is before their name, 1 is after<br>type: int  |
| ui_drivingCameraFreelook   | if true, driving cameras where there is no weapon defaults to freelook<br>type: bool<br>default: 0 (false)                 |
| ui_ignoreExplosiveWeapons  | if true, weapons marked as explosive will be ignored during auto-switches<br>type: bool<br>default: 1 (true)               |
| ui_name                    | player name<br>type: string  |
| ui_postArmFindBestWeapon   | if true, after arming players' best weapon will be selected<br>type: bool<br>default: 0 (false)                            |
| ui_rememberCameraMode      | use same camera mode as was previously used when re-entering a vehicle<br>type: bool<br>default: 0 (false)                 |
| ui_showGun                 | show gun<br>type: bool<br>default: 1 (true)  |
| wave_amplength             | Amplitude to length ratio<br>type: string (cheat)  |
| wave_angledeviation        | Angular deviation of waves<br>type: string (cheat)   |
| wave_fadespeed             | How quick individual waves fade in and out<br>type: string (cheat)   |
| wave_gravity               | Gavity constant for waves (controls speed based on wavelength)<br>type: string (cheat)                                     |
| wave_maxlength             | Maximum wave length<br>type: string (cheat)  |

| <b>cvar</b>                | <b>Description</b>  |
|----------------------------|---|
| wave_minlength             | Minimum wave length<br>type: string (cheat)   |
| win_allowMultipleInstances | allow multiple instances running concurrently<br>type: bool (cheat)<br>default: 0 (false)           |
| win_notaskkeys             | disable windows task keys<br>type: bool (cheat)<br>default: 1 (true)                                |
| win_outputDebugString      | type: bool (cheat)<br>default: 0 (false)  |
| win_outputEditString       | type: bool (cheat)<br>default: 1 (true)   |
| win_timerUpdate            | allows the game to be updated while dragging the window<br>type: bool (cheat)<br>default: 0 (false) |
| win_username               | windows user name<br>type: string (init)  |
| win_viewlog                | type: int (cheat)   |
| win_xpos                   | horizontal position of window<br>type: int  |
| win_ypos                   | vertical position of window<br>type: int  |

[ [Back to Enemy Territory Quake Wars](#) ] [ [Games database](#) ]

From:

<https://mwohlauer.d-n-s.name/wiki/> - [mwohlauer.d-n-s.name](https://mwohlauer.d-n-s.name/) / [www.mobile-infanterie.de](https://www.mobile-infanterie.de)

Permanent link:

[https://mwohlauer.d-n-s.name/wiki/doku.php?id=en:games:etqw:cvar\\_list&rev=1648848522](https://mwohlauer.d-n-s.name/wiki/doku.php?id=en:games:etqw:cvar_list&rev=1648848522)

Last update: **2022-04-01-21-28**

