

# Console Commands

## Variables

`audio_api <audioApi>`

**Type:** String

The underlying audio API to use.

`cache_version <value>`

**Type:** String

The engine version that the cache files are associated with. If this version doesn't match the current engine version, the cache will be cleared.

`cache_version_target <value>`

**Type:** String

If `cache_version` does not match this value, the cache files will be cleared and it will be set to it.

`cl_allowdownload <value>`

**Type:** String

`cl_allowupload <value>`

**Type:** String

`cl_audio_always_play <value>`

**Type:** String

0 = Don't play sounds if window isn't focused., 1 = Always play sounds

cl\_audio\_hrtf\_enabled <value>

**Type:** String

Enables or disables Head-related transfer function.

cl\_audio\_master\_volume <value>

**Type:** String

Volume scale for all sounds.

cl\_audio\_streaming\_enabled <value>

**Type:** String

0 = All sounds will be loaded immediately (= Slower load times), 1 = Some sounds will be loaded over time. (= Sounds might start with a delay)

cl\_controller\_axis\_input\_threshold <value>

**Type:** String

Axis inputs below this threshold will be ignored for regular console commands.

cl\_controller\_enabled <value>

**Type:** String

Enables or disables game controllers / joysticks.

cl\_downscale\_imported\_high\_resolution\_rma\_textures 1/0

**Type:** Boolean

If enabled, imported high-resolution RMA textures will be downscaled to a more memory-friendly size.

cl\_effects\_volume <value>

**Type:** String

Volume scale for effect sounds (e.g. footsteps, gunshots, explosions, etc.).

cl\_flex\_phoneme\_drag <value>

**Type:** String

Changes the time required for facial phoneme flexes to adapt to new phonemes.

**cl\_fov** <value>

**Type:** String

Horizontal world field of view.

**cl\_fov\_viewmodel** <value>

**Type:** String

Horizontal view model field of view.

**cl\_fps\_decay\_factor** <value>

**Type:** String

How slowly to decay the previous fps.

**cl\_gpu\_device** <value>

**Type:** String

The unique identifier for the GPU vendor and device to use. If empty (or invalid), the first detected device will be used. Usage: <vendorId,deviceId>

**cl\_gpu\_timer\_queries\_enabled** <value>

**Type:** String

Enables or disables GPU timer queries for debugging.

**cl\_gui\_volume** <value>

**Type:** String

Volume scale for sounds emitted by GUI elements.

**cl\_language** <value>

**Type:** String

Game language.

cl\_material\_streaming\_enabled <value>

**Type:** String

0 = All materials and textures will be loaded immediately (= Slower load times), 1 = All materials and textures will be loaded over time. (= Black textures until loaded)

cl\_max\_fps <value>

**Type:** String

FPS will be clamped at this value. A value of < 0 deactivates the limit.

cl\_mouse\_acceleration <value>

**Type:** String

Amount of mouse acceleration. 1 = No acceleration.

cl\_mouse\_pitch <value>

**Type:** String

Mouse sensitivity on the pitch axis. Use positive values to invert the axis.

cl\_mouse\_sensitivity <value>

**Type:** String

Scale for the mouse sensitivity.

cl\_mouse\_yaw <value>

**Type:** String

Mouse sensitivity on the yaw axis.

cl\_music\_volume <value>

**Type:** String

Volume scale for music.

cl\_physics\_simulation\_enabled <value>

**Type:** String

Enables or disables physics simulation.

`cl_port_tcp <value>`

**Type:** String

Port used for TCP transmissions.

`cl_port_udp <value>`

**Type:** String

Port used for UDP transmissions.

`cl_render_anti_aliasing <value>`

**Type:** String

0 = No Anti Aliasing, 1 = MSAA, 2 = FXAA.

`cl_render_brightness <value>`

**Type:** String

Scene brightness.

`cl_render_contrast <value>`

**Type:** String

Scene contrast.

`cl_render_depth_of_field <value>`

**Type:** String

Depth of field scale.

`cl_render_fov <value>`

**Type:** String

Specifies the horizontal field of view.

cl\_render\_frustum\_culling\_enabled <value>

**Type:** String

Enables or disables frustum culling.

cl\_render\_fxaa\_edge\_threshold <value>

**Type:** String

The minimum amount of local contrast required to apply algorithm.

cl\_render\_fxaa\_min\_edge\_threshold <value>

**Type:** String

Trims the algorithm from processing darks.

cl\_render\_fxaa\_sub\_pixel\_aliasing\_removal\_amount <value>

**Type:** String

The amount of sub-pixel aliasing removal.

cl\_render\_lod\_bias <value>

**Type:** String

Model LOD-bias. Higher values means higher LOD-Models will be used at lower distance (=Lower quality).

cl\_render\_monitor <value>

**Type:** String

Which monitor to use in fullscreen mode.

cl\_render\_motion\_blur <value>

**Type:** String

Amount of motion blur. 0 = disabled.

cl\_render\_msaa\_samples <value>

**Type:** String

1 = MSAAx2, 2 = MSAAx4, 3 = MSAAx8, etc.

cl\_render\_occlusion\_culling <value>

**Type:** String

0 = Off, 1 = Brute-Force, 2 = CHC++, 3 = Octree, 4 = BSP +Octree

cl\_render\_particle\_quality <value>

**Type:** String

Quality of particle systems. 0 = No particles will be rendered, 1 = Particles will be unlit, 2 = Particles will receive lighting, 3 = Particles will cast shadows.

cl\_render\_present\_mode <value>

**Type:** String

0 = Immediate, 1 = FIFO, 2 = Mailbox

cl\_render\_preset <value>

**Type:** String

Selected preset in display options.

cl\_render\_reflection\_quality <value>

**Type:** String

Changes the quality of reflections. 0 = Only static reflections, 1 = Dynamic reflections, 2 = Reflections with particle effects.

cl\_render\_resolution <value>

**Type:** String

Changes the internal rendering resolution. If left empty, the resolution will be the same as the window resolution!

cl\_render\_shader\_quality <value>

**Type:** String

Shader quality. The actual effect depends on the shader. 1 = Lowest Quality, 10 = Highest Quality

`cl_render_shadow_dynamic <value>`

**Type:** String

Turns dynamic shadows on or off.

`cl_render_shadow_lod_bias <value>`

**Type:** String

Model LOD-bias for shadows. Higher values means higher LOD-Models will be used at lower distance (=Lower quality). This value is added on top of `cl_render_lod_bias`!

`cl_render_shadow_pssm_split_count <value>`

**Type:** String

The number of cascades to be used for PSSM. Cannot be 0 or higher than 4.

`cl_render_shadow_pssm_update_frequency_offset <value>`

**Type:** String

Update frequency for PSSM shadows in frames, relative to '`cl_render_shadow_update_frequency`'.

`cl_render_shadow_resolution <value>`

**Type:** String

Shadowmap Resolution. Higher resolution equals higher quality shadows, but is also more expensive to render.

`cl_render_shadow_update_frequency <value>`

**Type:** String

Update frequency in frames. 0 = Updates every frame, 1 = Updates every second frame, etc.

`cl_render_ssao <value>`

**Type:** String

1 = Screen space ambient occlusion enabled, 0 = disabled.



cl\_render\_texture\_filtering <value>

**Type:** String

0 = Nearest Neighbour, 1 = Bilinear Filtering, 2 = Trilinear Filtering, 3 = Anisotropic Filtering x2, 4 = Anisotropic Filtering x4, 5 = Anisotropic Filtering x8, 6 = Anisotropic Filtering x16

cl\_render\_texture\_quality <value>

**Type:** String

0 = Minimal, 1 = Low, 2 = Medium, 3 = High, 4 = Very High.

cl\_render\_tone\_mapping <value>

**Type:** String

Changes the tone-mapping algorithm to use. -1 = Use the default, 0 = Gamma correction only, 1 = Reinhard, 2 = Hejil-Richard, 3 = Uncharted, 4 = Aces, 5 = Gran Turismo

cl\_render\_vr\_enabled <value>

**Type:** String

Turns support for virtual reality on or off.

cl\_render\_vsync\_enabled <value>

**Type:** String

Turns vertical sync on or off.

cl\_render\_window\_mode <value>

**Type:** String

0 = Fullscreen, 1 = Windowed, 2 = Windowed no-border.

cl\_show\_fps <value>

**Type:** String

Displays the current fps at the top left of the screen if enabled.

cl\_steam\_audio\_ambisonics\_order <value>

**Type:** String

The amount of directional detail in the simulation results. Range: [0,3].

cl\_steam\_audio\_enabled <value>

**Type:** String

Enables or disables steam audio.

cl\_steam\_audio\_ir\_duration <value>

**Type:** String

The time delay between a sound being emitted and the last audible reflection. Range: [0.5,4.0].

cl\_steam\_audio\_number\_of\_bounces <value>

**Type:** String

The maximum number of times any ray can bounce within the scene. Range: [1,32].

cl\_steam\_audio\_number\_of\_diffuse\_samples <value>

**Type:** String

The number of directions to consider when a ray bounces off a diffuse (or partly diffuse) surface. Range: [32,4096].

cl\_steam\_audio\_number\_of\_rays <value>

**Type:** String

The number of rays to trace from the listener. Range: [1024,131072].

cl\_steam\_audio\_propagation\_delay\_enabled <value>

**Type:** String

Enables or disables propagation delay if steam audio is enabled.

cl\_steam\_audio\_reverb\_enabled <value>

**Type:** String

Enables the reverb DSP effect.

cl\_steam\_audio\_reverb\_indirect\_binaural <value>

**Type:** String

Spatialize reverb using HRTF.

cl\_steam\_audio\_reverb\_simulation\_type <value>

**Type:** String

Real-time or baked. 0 = Real-time, 1 = Baked.

cl\_steam\_audio\_spatialize\_air\_absorption <value>

**Type:** String

Enable air absorption.

cl\_steam\_audio\_spatialize\_direct\_binaural <value>

**Type:** String

Spatialize direct sound using HRTF.

cl\_steam\_audio\_spatialize\_direct\_level <value>

**Type:** String

Relative level of direct sound. Range: [0,1].

cl\_steam\_audio\_spatialize\_distance\_attenuation <value>

**Type:** String

Enable distance attenuation. If disabled, regular distance attenuation will be used instead.

cl\_steam\_audio\_spatialize\_enabled <value>

**Type:** String

Enables the spatialize DSP effect.

cl\_steam\_audio\_spatialize\_hrtf\_interpolation <value>

**Type:** String

HRTF interpolation. 0 = Nearest, 1 = Bilinear.

`cl_steam_audio_spatialize_indirect <value>`

**Type:** String

Enable indirect sound.

`cl_steam_audio_spatialize_indirect_binaural <value>`

**Type:** String

Spatialize indirect sound using HRTF.

`cl_steam_audio_spatialize_indirect_level <value>`

**Type:** String

Relative level of indirect sound. Range: [0,10].

`cl_steam_audio_spatialize_occlusion_method <value>`

**Type:** String

Direct occlusion algorithm. 0 = Raycast, 1 = Partial.

`cl_steam_audio_spatialize_occlusion_mode <value>`

**Type:** String

Direct occlusion and transmission mode. 0 = Off, 1 = On, No Transmission, 2 = On, Frequency Independent Transmission, 3 = On, Frequency Dependent Transmission.

`cl_steam_audio_spatialize_simulation_type <value>`

**Type:** String

Real-time or baked. 0 = Real-time, 1 = Baked

`cl_steam_audio_spatialize_static_listener <value>`

**Type:** String

Uses static listener.

`cl_updaterate <value>`

**Type:** String

The amount of times per second user input is being transmitted to the server.

`cl_voice_volume <value>`

**Type:** String

Volume scale for voice (Coming from players and NPCs).

`cl_vr_hmd_view_enabled <value>`

**Type:** String

Enables or disables the view for virtual reality (Only works if virtual reality has been enabled in the menu.).

`cl_vr_mirror_window_enabled <value>`

**Type:** String

Enables or disables the mirror window for virtual reality (Only works if virtual reality has been enabled in the menu.).

`cl_water_surface_simulation_edge_iteration_count <value>`

**Type:** String

The more iterations, the more detailed the water simulation will be, but at a great performance cost.

`cl_water_surface_simulation_enable_gpu_acceleration <value>`

**Type:** String

If enabled, water surfaces will be simulated on the GPU instead of the CPU. This should greatly improve performance in most cases.

`cl_water_surface_simulation_spacing <value>`

**Type:** String

Grid spacing for simulating water surfaces. Lower values will result in a more detailed simulation, but at a great performance cost.

`cl_window_resolution <value>`

**Type:** String

Changes the window resolution.

**debug\_ai\_navigation** <value>

**Type:** String

Displays the current navigation path for all NPCs.

**debug\_bsp\_lock** <value>

**Type:** String

Locks BSP culling so it doesn't change when the camera moves.

**debug\_bsp\_nodes** <value>

**Type:** String

Prints and draws various debug information about the binary space partitioning of the level (if available). 1 = Draw leaf of camera location and all visible leaves, 2 = Same, but doesn't update continuously.

**debug\_hide\_gui** 1/0

**Type:** Boolean

Disables GUI rendering.

**debug\_nav\_show\_meshes** <value>

**Type:** String

Shows the navigation mesh for this map.

**debug\_occlusion\_culling\_freeze\_camera** <value>

**Type:** String

Freezes the current camera position in place for occlusion culling.

**debug\_particle\_blob\_show\_neighbor\_links** <value>

**Type:** String

Displays the links to adjacent neighbors for blob particles.

**debug\_physics\_draw** <value>

**Type:** String

Draws various physics information as wireframe.

**debug\_profiling\_enabled** <value>

**Type:** String

Enables profiling timers.

**debug\_render\_lock\_render\_command\_buffers** <value>

**Type:** String

Locks the current render command buffers and causes them to not be re-created every frame.

**debug\_render\_lock\_render\_queues** <value>

**Type:** String

Locks the current render queues and effectively disables the render queue thread.

**debug\_render\_normals** <value>

**Type:** String

Displays all normals as lines in the world. 0 = Disabled, 1 = Display normals, 2 = Display normals, tangents and bitangents, 3 = Display face normals, 4 = Display tangents, 5 = Display bitangents.

**debug\_render\_octree\_dynamic\_draw** <value>

**Type:** String

Draws the octree for dynamic objects.

**debug\_render\_octree\_static\_draw** <value>

**Type:** String

Draws the octree for static world geometry.

**debug\_steam\_audio\_probe\_boxes** <value>

**Type:** String

Displays the sound probe spheres for the current map.

host\_timescale <value>

**Type:** String

Specifies the timescale. A value of 2 means the game will run at twice the default speed.

log\_enabled <value>

**Type:** String

0 = Log disabled; 1 = Log errors only; 2 = Log errors and warnings; 3 = Log all console output

log\_file <value>

**Type:** String

The log-file the console output will be logged to.

lua\_open\_editor\_on\_error <value>

**Type:** String

1 = Whenever there's a Lua error, the engine will attempt to automatically open a Lua IDE and open the file and line which caused the error.

net\_graph <value>

**Type:** String

Displays a graph about current network transmissions.

net\_library <netLibrary>

**Type:** String

The underlying networking library to use for multiplayer games.

password <value>

**Type:** String

Password which will be used for the next connection attempt.



phys\_engine <physEngie>

**Type:** String

The underlying physics engine to use.

playername <value>

**Type:** String

Local player name.

rcon\_password <value>

**Type:** String

Specifies a password which can be used to run console commands remotely on a server. If no password is specified, this feature is disabled.

render\_api <renderApi>

**Type:** String

The underlying rendering API to use.

render\_bloom\_amount <value>

**Type:** String

Number of render passes to use for bloom.

render\_bloom\_enabled <value>

**Type:** String

Enables or disables bloom and glow effects.

render\_bloom\_resolution <value>

**Type:** String

The width for the bloom texture. The height will be calculated using the aspect ratio of the renderer.

render\_clear\_scene <value>

**Type:** String

0 = Default, 1 = Screen is cleared before scene is rendered.

`render_clear_scene_color <value>`

**Type:** String

Clear color to use if `render_clear_scene` is set to 1.

`render_csm_max_distance <value>`

**Type:** String

The maximum distance at which csm shadows are casted. A larger distance means lower quality.

`render_debug_mode 1/0`

**Type:** Boolean

0 = Disabled, 1 = Ambient Occlusion, 2 = Albedo Colors, 3 = Metalness, 4 = Roughness, 5 = Diffuse Lighting, 6 = Normals, 7 = Normal Map, 8 = Reflectance, 9 = IBL Prefilter, 10 = IBL Irradiance, 11 = Emission, 12 = Lightmaps, 13 = Lightmap Uvs, 14 = Unlit, 15 = Show CSM cascades, 16 = Shadow Map Depth, 17 = Forward+ Heatmap.

`render_draw_dynamic <value>`

**Type:** String

1 = Default, 0 = Don't draw props.

`render_draw_glow <value>`

**Type:** String

1 = Default, 0 = Glow-effects aren't drawn.

`render_draw_particles <value>`

**Type:** String

1 = Default, 0 = Particles aren't drawn.

`render_draw_scene <value>`

**Type:** String

1 = Default, 0 = Scene isn't drawn.

**render\_draw\_sky** <value>

**Type:** String

1 = Default, 0 = Sky isn't drawn.

**render\_draw\_static** <value>

**Type:** String

1 = Default, 0 = Don't draw level geometry.

**render\_draw\_translucent** <value>

**Type:** String

1 = Default, 0 = Don't draw translucent objects.

**render\_draw\_view** <value>

**Type:** String

1 = Default, 0 = View-Models aren't drawn.

**render\_draw\_water** <value>

**Type:** String

1 = Default, 0 = Water isn't drawn.

**render\_draw\_world** <value>

**Type:** String

1 = Default, 2 = Wireframe, 0 = Don't draw world.

**render\_dynamic\_lighting\_enabled** 1/0

**Type:** Boolean

Enables or disables dynamic lighting.

**render\_dynamic\_shadows\_enabled** 1/0

**Type:** Boolean

Enables or disables dynamic shadows.

`render_forwardplus_tile_size <value>`

**Type:** String

Changes the tile size used for the light culling in forward+.

`render_hdr_max_exposure <value>`

**Type:** String

Exposure will be clamped at this value. Can be overwritten by the map.

`render_ibl_enabled 1/0`

**Type:** Boolean

Enables or disables image-based lighting.

`render_instancing_enabled 1/0`

**Type:** Boolean

Enables or disables instanced rendering.

`render_instancing_threshold [2,inf]`

**Type:** UInt32

The threshold at which to start instancing entities if instanced rendering is enabled (`render_instancing_enabled`). Must not be lower than 2!

`render_multithreaded_rendering_enabled 1/0`

**Type:** Boolean

Enables or disables multi-threaded rendering. Some renderers (like OpenGL) don't support multi-threaded rendering and will ignore this flag.

`render_normalmapping_enabled <value>`

**Type:** String

1 = Normal mapping enabled, 0 = disabled.

**render\_pp\_nightvision** <value>

**Type:** String

Enables or disables night vision.

**render\_pp\_nightvision\_color\_amplification** <value>

**Type:** String

Amplification factor for pixels with low luminance.

**render\_pp\_nightvision\_luminance\_threshold** <value>

**Type:** String

Changes the luminance threshold for the night vision post-processing effect.

**render\_pp\_nightvision\_noise\_texture** <value>

**Type:** String

Noise texture to use for the night vision post-processing effect.

**render\_queue\_entities\_per\_worker\_job** [1,50]

**Type:** UInt32

Number of entities for each job processed by a worker thread.

**render\_queue\_worker\_jobs\_per\_batch** [0,10]

**Type:** UInt32

Number of worker jobs to accumulate in a batch before assigning a worker.

**render\_queue\_worker\_thread\_count** [1,10]

**Type:** UInt32

Number of threads to use for generating render queues.

**render\_shadow\_quality** <value>

**Type:** String

Shadowmap Quality. This affects the detail of the object shadows that are being rendered

render\_technique <value>

**Type:** String

0 = Rasterization, 1 = Raytracing

render\_texture\_filtering <value>

**Type:** String

0 = Nearest-Neighbor, 1 = Bilinear, 2 = trilinear.

render\_vsync\_enabled <value>

**Type:** String

1 = Vertical sync enabled, 0 = disabled.

sh\_lua\_remote\_debugging <value>

**Type:** String

0 = Remote debugging is disabled; 1 = Remote debugging is enabled serverside; 2 = Remote debugging is enabled clientside. Cannot be changed during an active game. Also requires the "-luaext" launch parameter. Remote debugging cannot be enabled clientside and serverside at the same time.

sh\_mount\_external\_game\_resources <value>

**Type:** String

If set to 1, the game will attempt to load missing resources from external games.

sky\_override <value>

**Type:** String

Replaces the material for all current skyboxes with the specified material.

steam\_steamworks\_enabled <value>

**Type:** String

Enables or disables steamworks.

## sv\_acceleration <value>

**Type:** String

Player acceleration. If this is too low, the player will be unable to reach full movement speed due to friction forces.

## sv\_allowcslua <value>

**Type:** String

1 = Clients can run own clientside scripts, 0 = Custom clientside scripts aren't allowed.

## sv\_allowdownload <value>

**Type:** String

Specifies whether clients are allowed to download resources from the server.

## sv\_allowupload <value>

**Type:** String

Specifies whether clients are allowed to upload resources to the server (e.g. spraylogos).

## sv\_cheats <value>

**Type:** String

Allows clients to run console commands which are marked as cheats if set to 1.

## sv\_debug\_physics\_draw <value>

**Type:** String

Draws various serverside physics information as wireframe.

## sv\_friction <value>

**Type:** String

Global friction scale.

## sv\_gamemode <value>

**Type:** String

Gamemode which should be used when starting a game.

**sv\_gravity** <value>

**Type:** String

Specifies the default gravity force and direction. Certain entities may overwrite this for themselves.

**sv\_maxplayers** <value>

**Type:** String

Specifies the maximum amount of players that are allowed to join the server.

**sv\_noclip\_speed** <value>

**Type:** String

The speed at which players can move in noclip mode.

**sv\_password** <value>

**Type:** String

Sets a password for the server. No player will be able to join, unless they provide the correct password using the 'password' command.

**sv\_physics\_simulation\_enabled** <value>

**Type:** String

Enables or disables physics simulation.

**sv\_player\_air\_move\_scale** <value>

**Type:** String

The scale at which players are still able to move while in the air. 0 = Players can't move until they've landed, 1 = Players can move freely while in the air

**sv\_port\_tcp** <value>

**Type:** String



TCP port which will be used when starting a server.

**sv\_port\_udp** <value>

**Type:** String

UDP port which will be used when starting a server.

**sv\_require\_authentication** 1/0

**Type:** Boolean

If enabled, clients will have to authenticate via steam to join the server.

**sv\_restitution** <value>

**Type:** String

Global restitution scale.

**sv\_servername** <value>

**Type:** String

The server name which will be displayed in the server browser.

**sv\_tickrate** <value>

**Type:** String

Specifies the tickrate. A higher tickrate means smoother and more reliable physics, but also more data to transmit to clients. Higher values can result in more lag for clients.

**sv\_timeout\_duration** <value>

**Type:** String

Amount of time until a client is forcibly dropped if no data has been received.

**sv\_use\_p2p\_if\_available** <value>

**Type:** String

Use a peer-to-peer connection if the selected networking layer supports it.

**sv\_water\_surface\_simulation\_edge\_iteration\_count** <value>

**Type:** String

The more iterations, the more detailed the water simulation will be, but at a great performance cost.

`sv_water_surface_simulation_shared <value>`

**Type:** String

If enabled, water surface simulation will be shared between client and server (Simulation is only performed once). This will only have an effect in single-player or on listen servers.

# Commands

`+attack`

Triggers the primary fire for the player character's equipped weapon.

`+attack2`

Triggers the secondary fire for the player character's equipped weapon.

`+attack3`

Triggers the tertiary fire for the player character's equipped weapon.

`+attack4`

Triggers the quaternary fire for the player character's equipped weapon.

`+backward`

Moves the player character backward while active.

`+crouch`

Prompts the player character to crouch.

`+forward`

Moves the player character forward while active.

`+jump`

Prompts the player character to jump.

## **+left**

Moves the player character left while active.

## **+reload**

Triggers the reload for the player character's equipped weapon.

## **+right**

Moves the player character right while active.

## **+sprint**

Enables sprinting while moving forward.

## **+turn\_down**

Continuously turns the player character leftwards.

## **+turn\_left**

Continuously turns the player character leftwards.

## **+turn\_right**

Continuously turns the player character leftwards.

## **+turn\_up**

Continuously turns the player character leftwards.

## **+use**

Prompts the player character to use whatever is in front of him.

## **+walk**

Enables walking while moving forward.

## **-attack**

Triggers the primary fire for the player character's equipped weapon.

## **-attack2**

Triggers the secondary fire for the player character's equipped weapon.

## **-attack3**

Triggers the tertiary fire for the player character's equipped weapon.

## **-attack4**

Triggers the quaternary fire for the player character's equipped weapon.

## **-backward**

Moves the player character backward while active.

## **-crouch**

Prompts the player character to crouch.

## **-forward**

Moves the player character forward while active.

## **-jump**

Prompts the player character to jump.

## **-left**

Moves the player character left while active.

## **-reload**

Triggers the reload for the player character's equipped weapon.

## **-right**

Moves the player character right while active.

## **-sprint**

Enables sprinting while moving forward.

## **-turn\_down**

Continuously turns the player character leftwards.

## **-turn\_left**

Continuously turns the player character leftwards.

## **-turn\_right**

Continuously turns the player character leftwards.

## **-turn\_up**

Continuously turns the player character leftwards.

## **-use**

Prompts the player character to use whatever is in front of him.

## **-walk**

Enables walking while moving forward.

## **asset\_clear\_unused**

Clears all unused assets from memory.

## **asset\_clear\_unused\_materials**

Clears all unused materials from memory.

## **asset\_clear\_unused\_models**

Clears all unused models from memory.

## **asset\_clear\_unused\_textures**

Clears all unused textures from memory.

## **bind**

Binds a key to a command string. Usage: bind <key> <command>

## **bind\_keys**

Prints a list of all bindable keys to the console.

## cl\_debug\_netmessages

Prints out debug information about recent net-messages.

## cl\_entities

Prints a list of all current clientside entities in the world.

## cl\_find

Finds similar console commands to whatever was given as argument.

## cl\_gpu\_timer\_queries\_dump

Prints all timer query results to the console.

## cl\_list

Prints a list of all clientside console commands to the console.

## cl\_send

Sends a text message to the server and displays it in the console. Usage: cl\_send <message>

## cl\_send\_udp

Sends a text message to the server via UDP and displays it in the console. Usage: cl\_send\_udp <message>

## cl\_steam\_audio\_reload\_scene

Reloads the steam audio scene cache.

## clear

Clears everything in the console.

## clear\_cache

Clears all of the cached engine files.

## closeserver

Closes the server (if active) and drops all connected clients.

## connect

Attempts to connect to the specified server. Usage: connect <ipV6/ipV4> or connect <steamId> or without arguments to re-try the last attempt.

## crash

Forces the engine to crash.

## credits

Prints a list of developers.

## debug\_ai\_schedule

Prints the current schedule behavior tree for the specified NPC on screen.

## debug\_ai\_schedule\_print

Prints the current schedule behavior tree for the specified NPC.

## debug\_aim\_info

Prints information about whatever the local player is looking at.

## debug\_audio\_aux\_effect

Applies a global DSP effect. Usage: debug\_audio\_aux\_effect <dspName> <gain>

## debug\_audio\_sounds

Prints information about all active server- and clientside sounds to the console.

## debug\_dump\_shader\_code

Dumps the glsl code for the specified shader.

## debug\_font

Displays the glyph map for the specified font. If no arguments are specified, all available fonts will be listed. Usage: debug\_font <fontName>

## debug\_font\_glyph\_map

Displays the glyph map for the specified font.

## debug\_glow\_bloom

Displays the scene glow texture on screen. Usage: debug\_glow\_bloom <1/0>

## debug\_gui\_cursor

Prints information about the GUI element currently hovered over by the cursor.

## debug\_hdr\_bloom

Displays the scene bloom texture on screen. Usage: debug\_hdr\_bloom <1/0>

## debug\_hitboxes

Displays the hitboxes for the entity under the crosshair, or the entity with the given name (If it was specified.).

## debug\_light\_shadowmap

Displays the depth map for the given light on screen. Call without arguments to turn the display off. Usage: debug\_light\_shadowmap <lightEntityIndex>

## debug\_light\_sources

Prints debug information about all light sources in the scene.

## debug\_lightmaps

Displays the lightmaps for the loaded level.

## debug\_nav\_path\_end

Sets an end point for a navigation path. Use debug\_nav\_path\_start to set the start point. The path will be drawn in the scene once both points have been set.

## debug\_nav\_path\_start

Sets a start point for a navigation path. Use debug\_nav\_path\_end to set the end point.

## debug\_particle\_alpha\_mode

Specifies the blend mode arguments for particle systems that use the 'custom' alpha mode.

Argument order: <srcColorBlendFactor> <dstColorBlendFactor> <srcAlphaBlendFactor> <dstAlphaBlendFactor> <opColor> <opAlpha>. Blend factor options: zero, one, src\_color, one\_minus\_src\_color, dst\_color, one\_minus\_dst\_color, src\_alpha, one\_minus\_src\_alpha, dst\_alpha, one\_minus\_dst\_alpha, constant\_color, one\_minus\_constant\_color, constant\_alpha,



one\_minus\_constant\_alpha, src\_alpha\_saturate, src1\_color, one\_minus\_src1\_color, src1\_alpha, one\_minus\_src1\_alpha Operation options: add, subtract, reverse\_subtract, min, max

## debug\_pbr\_ibl

Displays the irradiance, prefilter and brdf map for the closest cubemap.

## debug\_prepass

Displays prepass buffers to screen.

## debug\_profiling\_physics\_end

Prints physics profiling information for the last simulation step.

## debug\_profiling\_physics\_start

Prints physics profiling information for the last simulation step.

## debug\_profiling\_print

Prints the last profiled times.

## debug\_render\_depth\_buffer

Draws the scene depth buffer to screen.

## debug\_render\_info

Prints some timing information to the console.

## debug\_render\_memory\_budget

Prints information about the current GPU memory budget.

## debug\_render\_memory\_stats

Prints statistics about the current GPU memory usage.

## debug\_render\_octree\_dynamic\_find

Finds the specified entity in the octree for dynamic objects.

## debug\_render\_octree\_dynamic\_print

Prints the octree for dynamic objects to the console, or a file if a file name is specified.

## debug\_render\_octree\_static\_print

Prints the octree for static world geometry to the console, or a file if a file name is specified.

## debug\_render\_scene

Displays scene to screen.

## debug\_render\_stats

Prints information about the next frame.

## debug\_render\_validation\_error\_enabled

Enables or disables the specified validation error.

## debug\_ssao

Displays the ssao buffers to screen.

## debug\_steam\_audio\_dump\_scene

Saves the steam audio scene as OBJ file.

## debug\_texture\_mipmaps

Displays the mipmaps of the given texture. Usage: debug\_texture\_mipmaps <texturePath>

## debug\_textures

Prints information about the currently loaded textures.

## debug\_water

Displays the reflection, refraction and refraction depth map for the given water-entity on screen.

Call without arguments to turn the display off. Usage: debug\_light\_water <waterEntityIndex>

## disconnect

Disconnects from the server (if a connection is active), or closes the game if in single player mode.

## drop

Drops the player's active weapon.

## echo

Prints something to the console. Usage: echo <message>

## ent\_create

Creates and spawns a new entity with the given class name at the position in the world the player is looking at. Usage: ent\_create <className>

## ent\_input

Triggers the given input on the specified entity. Usage: ent\_input <entityName/entityClass> <input>

## ent\_remove

Removes the specified entity, or the entity the player is looking at if no argument is provided. Usage: ent\_remove <entityName/className>

## ent\_scale

Changes the scale of the specified entity. Usage: ent\_input <entityName/entityClass> <scale>

## entities

Prints a list of all current serverside entities in the world.

## exec

Executes a config file. Usage exec <fileName>

## exit

Exits the game.

## find

Finds similar console commands to whatever was given as argument.

## flashlight

Toggles the flashlight.

## fps

Prints the current fps to the console.

## getang

Prints the current angles of the local player to the console.

## getcamang

Prints the current angles of the camera to the console.

## getcampos

Prints the current position of the camera.

## getpos

Prints the current position of the local player to the console.

## give\_ammo

Gives the local player the specified ammunition.

## give\_weapon

Gives the local player the specified weapon and deploys it.

## godmode

Toggles god mode. Requires cheats to be enabled (Or single player mode running).

## heartbeat

Instantly sends a heartbeat to the master server.

## help

Find help about a convar/concommand.

## hurtme

Applies the specified amount of damage to self.

## keymappings

Prints a list of all active key bindings to the console.

## kick

Kicks the specified player for the given reason. Usage: kick <playerId/playerName> <reason>

## list

Prints a list of all serverside console commands to the console.

## list\_maps

Prints a list of all available list to the console.

## listmaps

## load

## loc\_reload

Reloads all localization files.

## log

Adds the specified message to the engine log. Usage: log <msg>.

## lua\_compile

Opens the specified lua-file and outputs a precompiled file with the same name (And the extension '.lua').

## lua\_exec

Opens and executes a lua-file on the server.

## lua\_exec\_cl

Opens and executes a lua-file on the client.

## lua\_help

Prints information about the specified function, library or enum (or the closest candiate). Usage: lua\_help <function/library/enum>.

## lua\_reload\_entities

Reloads the scripts for all registered lua entity classes.

## lua\_reload\_entity

Reloads the scripts for the given entity class. Usage: lua\_reload\_entity <className>

## lua\_reload\_weapon

Reloads the scripts for the given weapon class. Usage: lua\_reload\_weapon <className>

## lua\_reload\_weapons

Reloads the scripts for all registered lua weapon classes.

## lua\_run

Runs a lua command on the server lua state.

## lua\_run\_cl

Runs a lua command on the client lua state.

## lua\_run\_gui

Runs a lua command on the GUI lua state.

## map

Loads the given map immediately. Usage: map <mapName>

## map\_build\_reflection\_probes

Build all reflection probes in the map. Use the '-rebuild' argument to clear all current IBL textures first. Use 'debug\_pbr\_ibl' to check the probes after they have been built.

## map\_rebuild\_lightmaps

Rebuilds the lightmaps for the current map. Note that this will only work if the map was compiled with lightmap uvs.

## nav\_generate

Generates a navigation mesh for the current map and saves it as a navigation file.

## nav\_reload

Reloads the navigation mesh for the current map.

## next\_weapon

Switches to the next weapon in the player's inventory.

## noclip

Toggles noclip mode. Requires cheats to be enabled (Or single player mode running).

## notarget

Toggles notarget mode. Requires cheats to be enabled (Or single player mode running).

## previous\_weapon

Switches to the previous weapon in the player's inventory.

## quit

Exits the game.

## rcon

Sends the given command to the server and runs it serverside. Usage: rcon <command>

## reloadmaterial

Reloads the given material (and textures associated with it). Usage: reloadmaterial <materialFile>

## reloadmaterials

Reloads all loaded materials and textures.

## render\_api\_info

Prints information about the current render API to the console.

## save

## screenshot

Writes the contents of the screen into a .tga-file in 'screenshots/'. The name of the file will start with the map's name, or 'pragma' if no map is loaded. After the name follows an ID which is incremented each time. You can use 'screenshot 1' to take a screenshot without the GUI (Ingame only).

## setang

Changes the angles of the local player. Requires cheats to be enabled. Usage: setang <pitch> <yaw> <roll>

## setpos

Changes the position of the local player. Requires cheats to be enabled. Usage: setpos <x> <y> <z>

## shader\_list

Prints a list of all currently loaded shaders

## shader\_optimize

Uses LunarGLASS to optimize the specified shader.

## shader\_reload

Reloads a specific shader, or all registered shaders if no arguments are given. Usage: shader\_reload <shaderName>

## sound\_play

Precaches and immediately plays the given sound file. Usage: sound\_play <soundFile>

## sound\_stop

Stops all sounds that are currently playing.

## startserver

Starts an internet server. Requires a running game.

## status

Prints information about the current connection to the console.

## strip\_weapons



Strips all weapons from the local player.

## suicide

Commit suicide.

## sv\_debug\_netmessages

Prints out debug information about recent net-messages.

## sv\_send

Sends a text message to all connected clients and displays it in the console. Usage: sv\_send <message>

## sv\_send\_udp

Sends a text message to all connected clients via UDP and displays it in the console. Usage: sv\_send\_udp <message>

## thirdperson

Toggles between first- and third-person mode.

## toggle

Toggles the specified console variable between 0 and 1.

## toggleconsole

Toggles the developer console.

## udm\_convert

Converts a UDM file from binary to ASCII or the other way around.

## udm\_validate

Validates the specified UDM file.

## unbind

Unbinds the given key.

## unbindall

Unbinds all keys.

## version

Prints the current engine version to the console.

## vk\_dump\_extensions

Dumps all available vulkan extensions to 'vk\_extensions.txt'.

## vk\_dump\_features

Dumps all vulkan features to 'vk\_features.txt'.

## vk\_dump\_format\_properties

Dumps all vulkan format properties to 'vk\_format\_properties.txt'.

## vk\_dump\_image\_format\_properties

Dumps all vulkan image format properties to 'vk\_image\_format\_properties.txt'.

## vk\_dump\_layers

Dumps all available vulkan layers to 'vk\_layers.txt'.

## vk\_dump\_limits

Dumps all vulkan limitations to 'vk\_limits.txt'.

## vk\_dump\_memory\_stats

Dumps information about currently allocated GPU memory.

## vk\_print\_memory\_stats

Prints information about currently allocated GPU memory to the console.

---

Revision #1

Created 14 October 2021 12:38:43 by Silverlan

Updated 5 December 2022 09:45:21 by Silverlan