

(begin <expression(s)>)

returns the last expression in a sequence after evaluating the sequence in order.

(begin\_random <expression(s)>)

evaluates the sequence of expressions in random order and returns the last value evaluated.

(if <boolean> <then> [<else>])

returns one of two values based on the value of a condition.

(cond (<boolean1> <result1>) [(<boolean2> <result2>) [...]])

returns the value associated with the first true condition.

(set <variable name> <expression>)

set the value of a global variable.

(and <boolean(s)>)

returns true if all specified expressions are true.

(or <boolean(s)>)

returns true if any specified expressions are true.

(+ <number(s)>)

returns the sum of all specified expressions.

(- <number> <number>)

returns the difference of two expressions.

(\* <number(s)>)

returns the product of all specified expressions.

(/ <number> <number>)

returns the quotient of two expressions.

(min <number(s)>)

returns the minimum of all specified expressions.

(max <number(s)>)

returns the maximum of all specified expressions.

(= <expression> <expression>)

returns true if two expressions are equal

(!= <expression> <expression>)

returns true if two expressions are not equal

(> <number> <number>)

returns true if the first number is larger than the second.

(< <number> <number>)

returns true if the first number is smaller than the second.

(>= <number> <number>)

returns true if the first number is larger than or equal to the second.

(<= <number> <number>)

returns true if the first number is smaller than or equal to the second.

(sleep <short> [<script>])

pauses execution of this script (or, optionally, another script) for the specified number of ticks.

(sleep\_until <boolean> [<short>])

pauses execution of this script until the specified condition is true, checking once per second unless a different number of ticks is specified.

(wake <script name>)

wakes a sleeping script in the next update.

(inspect <expression>)

prints the value of an expression to the screen for debugging purposes.

(unit <object>)

converts an object to a unit.

(ai\_debug\_communication\_suppress <string(s)>)

suppresses (or stops suppressing) a set of AI communication types.

(ai\_debug\_communication\_ignore <string(s)>)

ignores (or stops ignoring) a set of AI communication types when printing out communications.

(ai\_debug\_communication\_focus <string(s)>)

focuses (or stops focusing) a set of unit vocalization types.

(not <boolean>)

returns the opposite of the expression.

(print <string>)

prints a string to the console.

(players)

returns a list of the players

(volume\_teleport\_players\_not\_inside <trigger\_volume> <cutscene\_flag>)

moves all players outside a specified trigger volume to a specified flag.

(volume\_test\_object <trigger\_volume> <object>)

returns true if the specified object is within the specified volume.

(volume\_test\_objects <trigger\_volume> <object\_list>)

returns true if any of the specified objects are within the specified volume.

(volume\_test\_objects\_all <trigger\_volume> <object\_list>)

returns true if any of the specified objects are within the specified volume.

(object\_teleport <object> <cutscene\_flag>)

moves the specified object to the specified flag.

(object\_set\_facing <object> <cutscene\_flag>)

turns the specified object in the direction of the specified flag.

(object\_set\_shield <object> <real>)

sets the shield vitality of the specified object (between 0 and 1).

(object\_set\_permutation <object> <string> <string>)

sets the desired region (use "" for all regions) to the permutation with the given name, e.g. (object\_set\_permutation flood "right arm" ~damaged)

(object\_create <object\_name>)

creates an object from the scenario.

(object\_destroy <object>)

destroys an object.

(object\_create\_anew <object\_name>)

creates an object, destroying it first if it already exists.

(object\_create\_containing <string>)

creates all objects from the scenario whose names contain the given substring.

(object\_create\_anew\_containing <string>)

creates anew all objects from the scenario whose names contain the given substring.

(object\_destroy\_containing <string>)

destroys all objects from the scenario whose names contain the given substring.

(object\_destroy\_all)

destroys all non player objects.

(list\_get <object\_list> <short>)

returns an item in an object list.

(list\_count <object\_list>)

returns the number of objects in a list

(effect\_new <effect> <cutscene\_flag>)

starts the specified effect at the specified flag.

(effect\_new\_on\_object\_marker <effect> <object> <string>)

starts the specified effect on the specified object at the specified marker.

(damage\_new <damage> <cutscene\_flag>)

causes the specified damage at the specified flag.

(damage\_object <damage> <object>)

causes the specified damage at the specified object.

(objects\_can\_see\_object <object\_list> <object> <real>)

returns true if any of the specified units are looking within the specified number of degrees of the object.

(objects\_can\_see\_flag <object\_list> <cutscene\_flag> <real>)

returns true if any of the specified units are looking within the specified number of degrees of the flag.

(objects\_delete\_by\_definition <object\_definition>)

deletes all objects of type <definition>

(sound\_set\_gain <string> <real>)

absolutely do not use this

(sound\_get\_gain <string>)

absolutely do not use this either

(script\_recompile)

recompiles scripts.

(script\_doc)

saves a file called hs\_doc.txt with parameters for all script commands.

(help <string>)

prints a description of the named function.

(random\_range <short> <short>)

returns a random value in the range [lower bound, upper bound)



(real\_random\_range <real> <real>)

returns a random value in the range [lower bound, upper bound)

(numeric\_countdown\_timer\_set <long> <boolean>)

<milliseconds>, <auto\_start>

(numeric\_countdown\_timer\_get <short>)

<digit\_index>

(numeric\_countdown\_timer\_stop)

(numeric\_countdown\_timer\_restart)

(breakable\_surfaces\_enable <boolean>)

enables or disables breakability of all breakable surfaces on level

(recording\_play <unit> <cutscene\_recording>)

make the specified unit run the specified cutscene recording.

(recording\_play\_and\_delete <unit> <cutscene\_recording>)

make the specified unit run the specified cutscene recording, deletes the unit when the animation finishes.

(recording\_play\_and\_hover <vehicle> <cutscene\_recording>)

make the specified vehicle run the specified cutscene recording, hovers the vehicle when the animation finishes.

(recording\_kill <unit>)

kill the specified unit's cutscene recording.

(recording\_time <unit>)

return the time remaining in the specified unit's cutscene recording.

(object\_set\_ranged\_attack\_inhibited <object> <boolean>)

FALSE prevents object from using ranged attack

(object\_set\_melee\_attack\_inhibited <object> <boolean>)

FALSE prevents object from using melee attack

(objects\_dump\_memory)

debugs object memory usage

(object\_set\_collideable <object> <boolean>)

FALSE prevents any object from colliding with the given object

(object\_set\_scale <object> <real> <short>)

sets the scale for a given object and interpolates over the given number of frames to achieve that scale

(objects\_attach <object> <string> <object> <string>)

attaches the second object to the first; both strings can be empty

(objects\_detach <object> <object>)

detaches from the given parent object the given child object

(garbage\_collect\_now)

causes all garbage objects except those visible to a player to be collected immediately

(object\_cannot\_take\_damage <object\_list>)

prevents an object from taking damage

(object\_can\_take\_damage <object\_list>)

allows an object to take damage again

(object\_beautify <object> <boolean>)

makes an object pretty for the remainder of the levels' cutscenes.

(objects\_predict <object\_list>)

loads textures necessary to draw a objects that are about to come on-screen.

(object\_type\_predict <object\_definition>)

loads textures necessary to draw an object that's about to come on-screen.

(object\_pvs\_activate <object>)

just another (old) name for object\_pvs\_set\_object.

(object\_pvs\_set\_object <object>)

sets the specified object as the special place that activates everything it sees.

(object\_pvs\_set\_camera <cutscene\_camera\_point>)

sets the specified cutscene camera point as the special place that activates everything it sees.

(object\_pvs\_clear)

removes the special place that activates everything it sees.

(render\_lights <boolean>)

enables/disables dynamic lights

(scenery\_get\_animation\_time <scenery>)

returns the number of ticks remaining in a custom animation (or zero, if the

animation is over).

(scenery\_animation\_start <scenery> <animation\_graph> <string>)

starts a custom animation playing on a piece of scenery

(scenery\_animation\_start\_at\_frame <scenery> <animation\_graph> <string>  
<short>)

starts a custom animation playing on a piece of scenery at a specific frame

(render\_effects <boolean>)

(unit\_can\_blink <unit> <boolean>)

allows a unit to blink or not (units never blink when they are dead)

(unit\_open <unit>)

opens the hatches on the given unit

(unit\_close <unit>)

closes the hatches on a given unit

(unit\_kill <unit>)

kills a given unit, no saving throw

(unit\_kill\_silent <unit>)

kills a given unit silently (doesn't make them play their normal death animation or sound)

(unit\_get\_custom\_animation\_time <unit>)

returns the number of ticks remaining in a unit's custom animation (or zero, if the animation is over).

(unit\_stop\_custom\_animation <unit>)

stops the custom animation running on the given unit.

(unit\_custom\_animation\_at\_frame <unit> <animation\_graph> <string> <boolean> <short>)

starts a custom animation playing on a unit at a specific frame index(interpolates into animation if next to last parameter is TRUE)

(custom\_animation <unit> <animation\_graph> <string> <boolean>)

starts a custom animation playing on a unit (interpolates into animation if last parameter is TRUE)

(custom\_animation\_list <object\_list> <animation\_graph> <string> <boolean>)

starts a custom animation playing on a unit list (interpolates into animation if last parameter is TRUE)

(unit\_is\_playing\_custom\_animation <unit>)

returns TRUE if the given unit is still playing a custom animation

(unit\_aim\_without\_turning <unit> <boolean>)

allows a unit to aim in place without turning

(unit\_set\_emotion <unit> <short>)

sets a unit's facial expression (-1 is none, other values depend on unit)

(unit\_set\_enterable\_by\_player <unit> <boolean>)

can be used to prevent the player from entering a vehicle

(unit\_enter\_vehicle <unit> <vehicle> <string>)

puts the specified unit in the specified vehicle (in the named seat)

(vehicle\_test\_seat\_list <vehicle> <string> <object\_list>)

tests whether the named seat has an object in the object list

(vehicle\_test\_seat <vehicle> <string> <unit>)

tests whether the named seat has a specified unit in it

(unit\_set\_emotion\_animation <unit> <string>)

sets the emotion animation to be used for the given unit

(unit\_exit\_vehicle <unit>)

makes a unit exit its vehicle

(unit\_set\_maximum\_vitality <unit> <real> <real>)

sets a unit's maximum body and shield vitality

(units\_set\_maximum\_vitality <object\_list> <real> <real>)

sets a group of units' maximum body and shield vitality

(unit\_set\_current\_vitality <unit> <real> <real>)

sets a unit's current body and shield vitality

(units\_set\_current\_vitality <object\_list> <real> <real>)

sets a group of units' current body and shield vitality

(vehicle\_load\_magic <unit> <string> <object\_list>)

makes a list of units (named or by encounter) magically get into a vehicle, in the substring-specified seats (e.g. CD-passenger... empty string matches all seats)

(vehicle\_unload <unit> <string>)

makes units get out of a vehicle from the substring-specified seats (e.g. CD-passenger... empty string matches all seats)

(magic\_seat\_name <string>)

all units controlled by the player will assume the given seat name (valid values



are 'asleep', 'alert', 'stand', 'crouch' and 'flee')

(unit\_set\_seat <unit> <string>)

this unit will assume the named seat

(magic\_melee\_attack)

causes player's unit to start a melee attack

(vehicle\_riders <unit>)

returns a list of all riders in a vehicle

(vehicle\_driver <unit>)

returns the driver of a vehicle

(vehicle\_gunner <unit>)

returns the gunner of a vehicle

(unit\_get\_health <unit>)

returns the health [0,1] of the unit, returns -1 if the unit does not exists

(unit\_get\_shield <unit>)

returns the shield [0,1] of the unit, returns -1 if the unit does not exists

(unit\_get\_total\_grenade\_count <unit>)

returns the total number of grenades for the given unit, 0 if it does not exist

(unit\_has\_weapon <unit> <object\_definition>)

returns TRUE if the <unit> has <object> as a weapon, FALSE otherwise

(unit\_has\_weapon\_readied <unit> <object\_definition>)

returns TRUE if the <unit> has <object> as the primary weapon, FALSE otherwise

(unit\_doesnt\_drop\_items <object\_list>)

prevents any of the given units from dropping weapons or grenades when they die

(unit\_impervious <object\_list> <boolean>)

prevents any of the given units from being knocked around or playing ping animations

(unit\_suspended <unit> <boolean>)

stops gravity from working on the given unit

(unit\_solo\_player\_integrated\_night\_vision\_is\_active)

returns whether the night-vision mode could be activated via the flashlight button

(units\_set\_desired\_flashlight\_state <object\_list> <boolean>)

sets the units' desired flashlight state

(unit\_set\_desired\_flashlight\_state <unit> <boolean>)

sets the unit's desired flashlight state

(unit\_get\_current\_flashlight\_state <unit>)

gets the unit's current flashlight state

(device\_set\_never\_appears\_locked <device> <boolean>)

changes a machine's never\_appears\_locked flag, but only if paul is a \*\*\*\*\*

(device\_get\_power <device>)

gets the current power of a named device

(device\_set\_power <device> <real>)

immediately sets the power of a named device to the given value

(device\_set\_position <device> <real>)

set the desired position of the given device (used for devices without explicit device groups)

(device\_get\_position <device>)

gets the current position of the given device (used for devices without explicit device groups)

(device\_set\_position\_immediate <device> <real>)

instantaneously changes the position of the given device (used for devices without explicit device groups)

(device\_group\_get <device\_group>)

returns the desired value of the specified device group.

(device\_group\_set <device\_group> <real>)

changes the desired value of the specified device group.

(device\_group\_set\_immediate <device\_group> <real>)

instantaneously changes the value of the specified device group.

(device\_one\_sided\_set <device> <boolean>)

TRUE makes the given device one-sided (only able to be opened from one direction), FALSE makes it two-sided

(device\_operates\_automatically\_set <device> <boolean>)

TRUE makes the given device open automatically when any biped is nearby, FALSE makes it not

(device\_group\_change\_only\_once\_more\_set <device\_group> <boolean>)

TRUE allows a device to change states only once

(breakable\_surfaces\_reset)

restores all breakable surfaces

(cheat\_all\_powerups)

drops all powerups near player

(cheat\_all\_weapons)

drops all weapons near player

(cheat\_spawn\_warthog)

drops a warthog near player

(cheat\_all\_vehicles)

drops all vehicles on player

(cheat\_teleport\_to\_camera)

teleports player to camera location

(cheat\_active\_camouflage)

gives the player active camouflage

(cheat\_active\_camouflage\_local\_player <short>)

gives the player active camouflage

(cheats\_load)

reloads the cheats.txt file

(ai\_free <ai>)

removes a group of actors from their encounter and sets them free

(ai\_free\_units <object\_list>)

removes a set of units from their encounter (if any) and sets them free

(ai\_attach <unit> <ai>)

attaches the specified unit to the specified encounter.

(ai\_attach\_free <unit> <actor\_variant>)

attaches a unit to a newly created free actor of the specified type

(ai\_detach <unit>)

detaches the specified unit from all AI.

(ai\_place <ai>)

places the specified encounter on the map.

(ai\_kill <ai>)

instantly kills the specified encounter and/or squad.

(ai\_kill\_silent <ai>)

instantly and silently (no animation or sound played) kills the specified encounter and/or squad.

(ai\_erase <ai>)

erases the specified encounter and/or squad.

(ai\_erase\_all)

erases all AI.

(ai\_select <ai>)

selects the specified encounter.

(ai\_deselect)

clears the selected encounter.

(ai\_spawn\_actor <ai>)

spawns a single actor in the specified encounter and/or squad.

(ai\_set\_respawn <ai> <boolean>)

enables or disables respawning in the specified encounter.

(ai\_set\_deaf <ai> <boolean>)

enables or disables hearing for actors in the specified encounter.

(ai\_set\_blind <ai> <boolean>)

enables or disables sight for actors in the specified encounter.

(ai\_magically\_see\_encounter <ai> <ai>)

makes one encounter magically aware of another.

(ai\_magically\_see\_players <ai>)

makes an encounter magically aware of nearby players.

(ai\_magically\_see\_unit <ai> <unit>)

makes an encounter magically aware of the specified unit.

(ai\_timer\_start <ai>)

makes a squad's delay timer start counting.

(ai\_timer\_expire <ai>)

makes a squad's delay timer expire and releases them to enter combat.

(ai\_attack <ai>)



makes the specified platoon(s) go into the attacking state.

(ai\_defend <ai>)

makes the specified platoon(s) go into the defending state.

(ai\_retreat <ai>)

makes all squads in the specified platoon(s) maneuver to their designated maneuver squads.

(ai\_maneuver <ai>)

makes all squads in the specified platoon(s) maneuver to their designated maneuver squads.

(ai\_maneuver\_enable <ai> <boolean>)

enables or disables the maneuver/retreat rule for an encounter or platoon. the rule will still trigger, but none of the actors will be given the order to change squads.

(ai\_migrate <ai> <ai>)

makes all or part of an encounter move to another encounter.

(ai\_migrate\_and\_speak <ai> <ai> <string>)

makes all or part of an encounter move to another encounter, and say their 'advance' or 'retreat' speech lines.

(ai\_migrate\_by\_unit <object\_list> <ai>)

makes a named vehicle or group of units move to another encounter.

(ai\_allegiance <team> <team>)

creates an allegiance between two teams.

(ai\_allegiance\_remove <team> <team>)

destroys an allegiance between two teams.

(ai\_living\_count <ai>)

return the number of living actors in the specified encounter and/or squad.

(ai\_living\_fraction <ai>)

return the fraction [0-1] of living actors in the specified encounter and/or squad.

(ai\_strength <ai>)

return the current strength (average body vitality from 0-1) of the specified encounter and/or squad.

(ai\_swarm\_count <ai>)

return the number of swarm actors in the specified encounter and/or squad.

(ai\_nonswarm\_count <ai>)

return the number of non-swarm actors in the specified encounter and/or squad.

(ai\_actors <ai>)

converts an ai reference to an object list.

(ai\_go\_to\_vehicle <ai> <unit> <string>)

tells a group of actors to get into a vehicle, in the substring-specified seats (e.g. passenger for pelican)... does not interrupt any actors who are already going to vehicles

(ai\_go\_to\_vehicle\_override <ai> <unit> <string>)

tells a group of actors to get into a vehicle, in the substring-specified seats (e.g. passenger for pelican)... NB: any actors who are already going to vehicles will stop and go to this one instead!

(ai\_going\_to\_vehicle <unit>)

return the number of actors that are still trying to get into the specified vehicle

(ai\_exit\_vehicle <ai>)

tells a group of actors to get out of any vehicles that they are in

(ai\_braindead <ai> <boolean>)

makes a group of actors braindead, or restores them to life (in their initial state)

(ai\_braindead\_by\_unit <object\_list> <boolean>)

makes a list of objects braindead, or restores them to life. if you pass in a vehicle

index, it makes all actors in that vehicle braindead (including any built-in guns)

(ai\_disregard <object\_list> <boolean>)

if TRUE, forces all actors to completely disregard the specified units, otherwise lets them acknowledge the units again

(ai\_prefer\_target <object\_list> <boolean>)

if TRUE, \*ALL\* enemies will prefer to attack the specified units. if FALSE, removes the preference.

(ai\_teleport\_to\_starting\_location <ai>)

teleports a group of actors to the starting locations of their current squad(s)

(ai\_teleport\_to\_starting\_location\_if\_unsupported <ai>)

teleports a group of actors to the starting locations of their current squad(s), only if they are not supported by solid ground (i.e. if they are falling after switching BSPs)

(ai\_renew <ai>)

refreshes the health and grenade count of a group of actors, so they are as good as new

(ai\_try\_to\_fight\_nothing <ai>)

removes the preferential target setting from a group of actors

(ai\_try\_to\_fight <ai> <ai>)

causes a group of actors to preferentially target another group of actors

(ai\_try\_to\_fight\_player <ai>)

causes a group of actors to preferentially target the player

(ai\_command\_list <ai> <ai\_command\_list>)

tells a group of actors to begin executing the specified command list

(ai\_command\_list\_by\_unit <unit> <ai\_command\_list>)

tells a named unit to begin executing the specified command list

(ai\_command\_list\_advance <ai>)

tells a group of actors that are running a command list that they may advance further along the list (if they are waiting for a stimulus)

(ai\_command\_list\_advance\_by\_unit <unit>)

just like ai\_command\_list\_advance but operates upon a unit instead

(ai\_command\_list\_status <object\_list>)

gets the status of a number of units running command lists: 0 = none, 1 = finished command list, 2 = waiting for stimulus, 3 = running command list

(ai\_is\_attacking <ai>)

returns whether a platoon is in the attacking mode (or if an encounter is

specified, returns whether any platoon in that encounter is attacking)

(ai\_force\_active <ai> <boolean>)

forces an encounter to remain active (i.e. not freeze in place) even if there are no players nearby

(ai\_force\_active\_by\_unit <unit> <boolean>)

forces a named actor that is NOT in an encounter to remain active (i.e. not freeze in place) even if there are no players nearby

(ai\_set\_return\_state <ai> <ai\_default\_state>)

sets the state that a group of actors will return to when they have nothing to do

(ai\_set\_current\_state <ai> <ai\_default\_state>)

sets the current state of a group of actors. WARNING: may have unpredictable results on actors that are in combat

(ai\_playfight <ai> <boolean>)

sets an encounter to be playfighting or not

(ai\_status <ai>)

returns the most severe combat status of a group of actors (0=inactive, 1=noncombat, 2=guarding, 3=search/suspicious, 4=definite enemy(heard or magic awareness), 5=visible enemy, 6=engaging in combat.

(ai\_reconnect)

reconnects all AI information to the current structure bsp (use this after you create encounters or command lists in sapien, or place new firing points or command list points)

(ai\_vehicle\_encounter <unit> <ai>)

sets a vehicle to 'belong' to a particular encounter/squad. any actors who get into the vehicle will be placed in this squad. NB: vehicles potentially drivable by multiple teams need their own encounter!

(ai\_vehicle\_enterable\_distance <unit> <real>)

sets a vehicle as being impulsively enterable for actors within a certain distance

(ai\_vehicle\_enterable\_team <unit> <team>)

sets a vehicle as being impulsively enterable for actors on a certain team

(ai\_vehicle\_enterable\_actor\_type <unit> <actor\_type>)

sets a vehicle as being impulsively enterable for actors of a certain type (grunt, elite, marine etc)

(ai\_vehicle\_enterable\_actors <unit> <ai>)

sets a vehicle as being impulsively enterable for a certain encounter/squad of actors

(ai\_vehicle\_enterable\_disable <unit>)

disables actors from impulsively getting into a vehicle (this is the default state for newly placed vehicles)

(ai\_look\_at\_object <unit> <object>)

tells an actor to look at an object until further notice

(ai\_stop\_looking <unit>)

tells an actor to stop looking at whatever it's looking at

(ai\_automatic\_migration\_target <ai> <boolean>)

enables or disables a squad as being an automatic migration target

(ai\_follow\_target\_disable <ai>)

turns off following for an encounter

(ai\_follow\_target\_players <ai>)

sets the follow target for an encounter to be the closest player

(ai\_follow\_target\_unit <ai> <unit>)

sets the follow target for an encounter to be a specific unit

(ai\_follow\_target\_ai <ai> <ai>)

sets the follow target for an encounter to be a group of AI (encounter, squad or platoon)

(ai\_follow\_distance <ai> <real>)



sets the distance threshold which will cause squads to migrate when following someone

(ai\_conversation <conversation>)

tries to add an entry to the list of conversations waiting to play. returns FALSE if the required units could not be found to play the conversation, or if the player is too far away and the 'delay' flag is not set.

(ai\_conversation\_stop <conversation>)

stops a conversation from playing or trying to play

(ai\_conversation\_advance <conversation>)

tells a conversation that it may advance

(ai\_conversation\_line <conversation>)

returns which line the conversation is currently playing, or 999 if the conversation is not currently playing

(ai\_conversation\_status <conversation>)

returns the status of a conversation (0=none, 1=trying to begin, 2=waiting for guys to get in position, 3=playing, 4=waiting to advance, 5=could not begin, 6=finished successfully, 7=aborted midway)

(ai\_link\_activation <ai> <ai>)

links the first encounter so that it will be made active whenever it detects that the second encounter is active

(ai\_berserk <ai> <boolean>)

forces a group of actors to start or stop berserking

(ai\_set\_team <ai> <team>)

makes an encounter change to a new team

(ai\_allow\_charge <ai> <boolean>)

either enables or disables charging behavior for a group of actors

(ai\_allow\_dormant <ai> <boolean>)

either enables or disables automatic dormancy for a group of actors

(ai\_allegiance\_broken <team> <team>)

returns whether two teams have an allegiance that is currently broken by traitorous behavior

(camera\_control <boolean>)

toggles script control of the camera.

(camera\_set <cutscene\_camera\_point> <short>)

moves the camera to the specified camera point over the specified number of ticks.

(camera\_set\_relative <cutscene\_camera\_point> <short> <object>)

moves the camera to the specified camera point over the specified number of ticks (position is relative to the specified object).

(camera\_set\_animation <animation\_graph> <string>)

begins a prerecorded camera animation.

(camera\_set\_first\_person <unit>)

makes the scripted camera follow a unit.

(camera\_set\_dead <unit>)

makes the scripted camera zoom out around a unit as if it were dead.

(camera\_time)

returns the number of ticks remaining in the current camera interpolation.

(debug\_camera\_load)

loads the saved camera position and facing.

(debug\_camera\_save)

saves the camera position and facing.

(game\_speed <real>)

changes the game speed.

(game\_time)

gets ticks elapsed since the start of the game.

(game\_variant <string>)

set the game engine

(game\_difficulty\_get)

returns the current difficulty setting, but lies to you and will never return easy, instead returning normal

(game\_difficulty\_get\_real)

returns the actual current difficulty setting without lying

(profile\_service\_clear\_timers)

clears the timers that are present in the profiling service

(profile\_service\_dump\_timers)

dumps the profiling service timers

(map\_reset)

starts the map from the beginning.

(map\_name <string>)

changes the name of the solo player map.

(multiplayer\_map\_name <string>)

changes the name of the multiplayer map

(game\_difficulty\_set <game\_difficulty>)

changes the difficulty setting for the next map to be loaded.

(crash <string>)

crashes (for debugging).

(switch\_bsp <short>)

takes off your condom and changes to a different structure bsp

(structure\_bsp\_index)

returns the current structure bsp index

(version)

prints the build version.

(playback)

starts game in film playback mode

(quit)

quits the game

(texture\_cache\_flush)

don't make me kick your ass

(sound\_cache\_flush)

i'm a rebel!

(sound\_cache\_dump\_to\_file)

dump dat \*\*\*\*!

(debug\_memory)

dumps memory leaks.

(debug\_memory\_by\_file)

dumps memory leaks by source file.

(debug\_memory\_for\_file <string>)

dumps memory leaks from the specified source file.

(debug\_tags)

writes all memory being used by tag files into tag\_dump.txt

(profile\_reset)

resets profiling data.

(profile\_dump <string>)

dumps profile based on a substring.

(profile\_activate <string>)

activates profile sections based on a substring.

(profile\_deactivate <string>)

deactivates profile sections based on a substring.

(profile\_graph\_toggle <string>)

enables or disables profile graph display of a particular value.

(debug\_pvs <boolean>)

displays the current pvs.

(radiosity\_start)

starts radiosity computation.

(radiosity\_save)

saves radiosity solution.

(radiosity\_debug\_point)

tests sun occlusion at a point.

(ai <boolean>)

turns all AI on or off.

(ai\_dialogue\_triggers <boolean>)

turns impromptu dialogue on or off.

(ai\_grenades <boolean>)

turns grenade inventory on or off.

(ai\_lines)

cycles through AI line-spray modes

(ai\_debug\_sound\_point\_set)

drops the AI debugging sound point at the camera location

(ai\_debug\_vocalize <string> <string>)

makes the selected AI vocalize



(ai\_debug\_teleport\_to <ai>)

teleports all players to the specified encounter

(ai\_debug\_speak <string>)

makes the currently selected AI speak a vocalization (e.g. ai\_speak "pain minor")

(ai\_debug\_speak\_list <string>)

makes the currently selected AI speak a list of vocalizations (e.g. ai\_speak\_list "involuntary")

(fade\_in <real> <real> <real> <short>)

does a screen fade in from a particular color

(fade\_out <real> <real> <real> <short>)

does a screen fade out to a particular color

(cinematic\_start)

initializes game to start a cinematic (interruptive) cutscene

(cinematic\_stop)

initializes the game to end a cinematic (interruptive) cutscene

(cinematic\_abort)

aborts a cinematic

(cinematic\_skip\_start\_internal)

(cinematic\_skip\_stop\_internal)

(cinematic\_show\_letterbox <boolean>)

sets or removes the letterbox bars

(cinematic\_set\_title <cutscene\_title>)

activates the chapter title

(cinematic\_set\_title\_delayed <cutscene\_title> <real>)

activates the chapter title, delayed by <real> seconds

(cinematic\_suppress\_bsp\_object\_creation <boolean>)

suppresses or enables the automatic creation of objects during cutscenes due to a bsp switch

(attract\_mode\_start)

(game\_won)

causes the player to successfully finish the current level and move to the next

(game\_lost)

causes the player to revert to his previous saved game

(game\_safe\_to\_save)

returns FALSE if it would be a bad idea to save the player's game right now

(game\_all\_quiet)

returns FALSE if there are bad guys around, projectiles in the air, etc.

(game\_safe\_to\_speak)

returns FALSE if it would be a bad idea to save the player's game right now

(game\_is\_cooperative)

returns TRUE if the game is cooperative

(game\_save)

checks to see if it is safe to save game, then saves (gives up after 8 seconds)

(game\_save\_cancel)

cancels any pending game\_save, timeout or not

(game\_save\_no\_timeout)

checks to see if it is safe to save game, then saves (this version never gives up)

(game\_save\_totally\_unsafe)

disregards player's current situation

(game\_saving)

checks to see if the game is trying to save the map.

(game\_revert)

reverts to last saved game, if any (for testing, the first \*\*\*\*\* that does this to me gets it in the head)

(game\_reverted)

don't use this for anything, you black-hearted \*\*\*\*\*s.

(core\_save)

saves debug game state to core\core.bin

(core\_save\_name <string>)

saves debug game state to core\<path>

(core\_load)

loads debug game state from core\core.bin

(core\_load\_at\_startup)

loads debug game state from core\core.bin as soon as the map is initialized

(core\_load\_name <string>)

loads debug game state from core\<path>

(core\_load\_name\_at\_startup <string>)

loads debug game state from core\<path> as soon as the map is initialized

(game\_skip\_ticks <short>)

skips <short> amount of game ticks. ONLY USE IN CUTSCENES!!!

(sound\_impulse\_predict <sound> <boolean>)

loads an impulse sound into the sound cache ready for playback.

(sound\_impulse\_start <sound> <object> <real>)

plays an impulse sound from the specified source object (or "none"), with the specified scale.

(sound\_impulse\_time <sound>)

returns the time remaining for the specified impulse sound.

(sound\_impulse\_stop <sound>)

stops the specified impulse sound.

(sound\_looping\_predict <looping\_sound>)

your mom.

(sound\_looping\_start <looping\_sound> <object> <real>)

plays a looping sound from the specified source object (or "none"), with the specified scale.

(sound\_looping\_stop <looping\_sound>)

stops the specified looping sound.

(sound\_looping\_set\_scale <looping\_sound> <real>)

changes the scale of the sound (which should affect the volume) within the range 0 to 1.

(sound\_looping\_set\_alternate <looping\_sound> <boolean>)

enables or disables the alternate loop/alternate end for a looping sound.

(debug\_sounds\_enable <string> <boolean>)

enables or disabled all sound classes matching the substring.

(debug\_sounds\_distances <string> <real> <real>)

changes the minimum and maximum distances for all sound classes matching the substring.

(debug\_sounds\_wet <string> <real>)

changes the reverb level for all sound classes matching the substring.

(sound\_enable <boolean>)

enables or disables all sound.

(sound\_set\_master\_gain <real>)

Set the game's master gain

(sound\_get\_master\_gain)

Returns the game's master gain

(sound\_set\_music\_gain <real>)

Set the game's music gain

(sound\_get\_music\_gain)

Returns the game's music gain

(sound\_set\_effects\_gain <real>)

Set the game's effects gain

(sound\_get\_effects\_gain)

Returns the game's effects gain

(sound\_class\_set\_gain <string> <real> <short>)

changes the gain on the specified sound class(es) to the specified game over the specified number of ticks.

(vehicle\_hover <vehicle> <boolean>)

stops the vehicle from running real physics and runs fake hovering physics instead.

(players\_unzoom\_all)

resets zoom levels on all players

(player\_enable\_input <boolean>)

toggle player input. the player can still free-look, but nothing else.

(player\_camera\_control <boolean>)

enables/disables camera control globally

(player\_action\_test\_reset)

resets the player action test state so that all tests will return false.



(player\_action\_test\_jump)

returns true if any player has jumped since the last call to (player\_action\_test\_reset).

(player\_action\_test\_primary\_trigger)

returns true if any player has used primary trigger since the last call to (player\_action\_test\_reset).

(player\_action\_test\_grenade\_trigger)

returns true if any player has used grenade trigger since the last call to (player\_action\_test\_reset).

(player\_action\_test\_zoom)

returns true if any player has hit the zoom button since the last call to (player\_action\_test\_reset).

(player\_action\_test\_action)

returns true if any player has hit the action key since the last call to (player\_action\_test\_reset).

(player\_action\_test\_accept)

returns true if any player has hit accept since the last call to (player\_action\_test\_reset).

(player\_action\_test\_back)

returns true if any player has hit the back key since the last call to (player\_action\_test\_reset).

(player\_action\_test\_look\_relative\_up)

returns true if any player has looked up since the last call to (player\_action\_test\_reset).

(player\_action\_test\_look\_relative\_down)

returns true if any player has looked down since the last call to (player\_action\_test\_reset).

(player\_action\_test\_look\_relative\_left)

returns true if any player has looked left since the last call to (player\_action\_test\_reset).

(player\_action\_test\_look\_relative\_right)

returns true if any player has looked right since the last call to (player\_action\_test\_reset).

(player\_action\_test\_look\_relative\_all\_directions)

returns true if any player has looked up, down, left, and right since the last call to (player\_action\_test\_reset).

(player\_action\_test\_move\_relative\_all\_directions)

returns true if any player has moved forward, backward, left, and right since the last call to (player\_action\_test\_reset).

(player\_add\_equipment <unit> <starting\_profile> <boolean>)

adds/resets the player's health, shield, and inventory (weapons and grenades) to the named profile. resets if third parameter is true, adds if false.

(debug\_teleport\_player <short> <short>)

(show\_hud <boolean>)

shows or hides the hud

(show\_hud\_help\_text <boolean>)

shows or hides the hud help text

(enable\_hud\_help\_flash <boolean>)

starts/stops the help text flashing

(hud\_help\_flash\_restart)

resets the timer for the help text flashing

(activate\_nav\_point\_flag <navpoint> <unit> <cutscene\_flag> <real>)

activates a nav point type <string> attached to (local) player <unit> anchored to a flag with a vertical offset <real>. If the player is not local to the machine, this will fail

(activate\_nav\_point\_object <navpoint> <unit> <object> <real>)

activates a nav point type <string> attached to (local) player <unit> anchored to an object with a vertical offset <real>. If the player is not local to the machine, this will fail

(activate\_team\_nav\_point\_flag <navpoint> <team> <cutscene\_flag> <real>)

activates a nav point type <string> attached to a team anchored to a flag with a vertical offset <real>. If the player is not local to the machine, this will fail

(activate\_team\_nav\_point\_object <navpoint> <team> <object> <real>)

activates a nav point type <string> attached to a team anchored to an object with a vertical offset <real>. If the player is not local to the machine, this will fail

(deactivate\_nav\_point\_flag <unit> <cutscene\_flag>)

deactivates a nav point type attached to a player <unit> anchored to a flag

(deactivate\_nav\_point\_object <unit> <object>)

deactivates a nav point type attached to a player <unit> anchored to an object

(deactivate\_team\_nav\_point\_flag <team> <cutscene\_flag>)

deactivates a nav point type attached to a team anchored to a flag

(deactivate\_team\_nav\_point\_object <team> <object>)

deactivates a nav point type attached to a team anchored to an object

(hud\_team\_icon\_set\_pos <long> <long>)

\*\*\*\*

(hud\_team\_icon\_set\_scale <real> <real>)

\*\*\*\*

(hud\_team\_background\_set\_pos <long> <long>)

\*\*\*\*

(hud\_team\_background\_set\_scale <real> <real>)

\*\*\*\*

(cls)

clears console text from the screen

(connect <string> <string>)

Attempt to connect to server - use ip:port password as parameters

(disconnect)

Disconnect from a server

(hammer\_begin <string> <string> <long> <short> <short>)

hammers the server by connecting and disconnecting repeatedly.

(hammer\_stop)

stops hammering the server.

(network\_server\_dump)

Dumps info on network server.

(network\_client\_dump)

Dumps info on network client.

(net\_graph\_clear)

Clears the net\_graph.

(net\_graph\_show <string> <string>)

Changes the net\_graph display (bytes/packets, sent/received)

(play\_update\_history <long> <boolean>)

Playback client input history starting from the specified last completed update id.

(show\_player\_update\_stats)

Shows update history playback stats.

(message\_metrics\_clear)

clears network messaging metrics

(message\_metrics\_dump <string>)

dumps network messaging metrics to given file ("" for default)

(error\_overflow\_suppression <boolean>)

enables or disables the suppression of error spamming

(structure\_lens\_flares\_place)

places lens flares in the structure bsp

(player\_effect\_set\_max\_translation <real> <real> <real>)

<x> <y> <z>

(player\_effect\_set\_max\_rotation <real> <real> <real>)

<yaw> <pitch> <roll>

(player\_effect\_set\_max\_vibrate <real> <real>)

<left> <right>

(player\_effect\_start <real> <real>)

<max\_intensity> <attack time>

(player\_effect\_stop <real>)

<decay>

(hud\_show\_health <boolean>)

hides/shows the health panel

(hud\_blink\_health <boolean>)

starts/stops manual blinking of the health panel

(hud\_show\_shield <boolean>)

hides/shows the shield panel

(hud\_blink\_shield <boolean>)

starts/stops manual blinking of the shield panel

(hud\_show\_motion\_sensor <boolean>)

hides/shows the motion sensor panel

(hud\_blink\_motion\_sensor <boolean>)

starts/stops manual blinking of the motion sensor panel

(hud\_show\_crosshair <boolean>)

hides/shows the weapon crosshair



(hud\_clear\_messages)

clears all non-state messages on the hud

(hud\_set\_help\_text <hud\_message>)

displays <message> as the help text

(hud\_set\_objective\_text <hud\_message>)

sets <message> as the current objective

(hud\_set\_timer\_time <short> <short>)

sets the time for the timer to <short> minutes and <short> seconds, and starts and displays timer

(hud\_set\_timer\_warning\_time <short> <short>)

sets the warning time for the timer to <short> minutes and <short> seconds

(hud\_set\_timer\_position <short> <short> <hud\_corner>)

sets the timer upper left position to (x, y)=(<short>, <short>)

(show\_hud\_timer <boolean>)

displays the hud timer

(pause\_hud\_timer <boolean>)

pauses or unpauses the hud timer

(hud\_get\_timer\_ticks)

returns the ticks left on the hud timer

(time\_code\_show <boolean>)

shows the time code timer

(time\_code\_start <boolean>)

starts/stops the time code timer

(time\_code\_reset)

resets the time code timer

(reload\_shader\_transparent\_chicago)

(rasterizer\_reload\_effects)

check for shader changes

(set\_gamma <long>)

set the gamma

(rasterizer\_fixed\_function\_ambient <long>)

set the ambient light value for fixed function

(rasterizer\_decals\_flush)

flush all decals

(rasterizer\_fps\_accumulate)

average fps

(rasterizer\_model\_ambient\_reflection\_tint <real> <real> <real> <real>)

(rasterizer\_lights\_reset\_for\_new\_map)

(script\_screen\_effect\_set\_value <short> <real>)

sets a screen effect script value

(cinematic\_screen\_effect\_start <boolean>)

starts screen effect; pass TRUE to clear

(cinematic\_screen\_effect\_set\_convolution <short> <short> <real> <real> <real>)

sets the convolution effect

(cinematic\_screen\_effect\_set\_filter <real> <real> <real> <real> <boolean>  
<real>)

sets the filter effect

(cinematic\_screen\_effect\_set\_filter\_desaturation\_tint <real> <real> <real>)

sets the desaturation filter tint color

(cinematic\_screen\_effect\_set\_video <short> <real>)

sets the video effect: <noise intensity[0,1]>, <overbright: 0=none, 1=2x, 2=4x>

(cinematic\_screen\_effect\_stop)

returns control of the screen effects to the rest of the game

(cinematic\_set\_near\_clip\_distance <real>)

(delete\_save\_game\_files)

delete all custom profile files

(fast\_setup\_network\_server <string> <string> <boolean>)

sets up a network server with the given map name, game variant, and true for remote connections, false for not

(profile\_unlock\_solo\_levels)

unlocks all the solo player levels for player 1's profile

(player0\_look\_invert\_pitch <boolean>)

invert player0's look

(player0\_look\_pitch\_is\_inverted)

returns TRUE if player0's look pitch is inverted

(player0\_joystick\_set\_is\_normal)

returns TRUE if player0 is using the normal joystick set

(ui\_widget\_show\_path <boolean>)

blah blah

(display\_scenario\_help <short>)

display in-game help dialog

(sound\_enable\_eax <boolean>)

Enable or disable EAX extensions

(sound\_eax\_enabled)

Returns true if EAX extensions are enabled

(sound\_set\_env <short>)

Change environment preset

(sound\_enable\_hardware <boolean> <boolean>)

Enable or disable hardware sound buffers

(sound\_set\_supplementary\_buffers <short> <boolean>)

Set the amount of supplementary buffers

(sound\_get\_supplementary\_buffers)

Get the amount of supplementary buffers

(sound\_set\_rolloff <real>)

Set the DSound rolloff value

(sound\_set\_factor <real>)

Set the DSound factor value

(input\_get\_joy\_count)

test function to return the number of joysticks enumerated

(input\_is\_joy\_active <short>)

test function to determine if an enumerated joystick is activated or not

(input\_activate\_joy <short> <short>)

activates an enumerated joystick into a logical joystick slot

(input\_deactivate\_joy <short>)

deactivates an enumerated joystick, freeing up the logical joystick slot

(input\_find\_joystick <string>)

test function to find a joystick by GUID (string representation)

(input\_show\_joystick\_info)

test function to show the enumerated joystick information for all joystick

(input\_find\_default <string>)

test function that looks up a default profile for a deviceid

(config\_one\_control <string>)

test function to configure a single control

(get\_yaw\_rate <short>)

gets the yaw rate for the given player number

(get\_pitch\_rate <short>)

gets the yaw rate for the given player number

(set\_yaw\_rate <short> <real>)

sets the yaw rate for the given player number

(set\_pitch\_rate <short> <real>)

sets the yaw rate for the given player number

(get\_digital\_forward\_throttle <short>)

gets the amount of forward throttle applied by digital device stimuli

(set\_digital\_forward\_throttle <short> <real>)

sets the amount of forward throttle applied by digital device stimuli

(get\_digital\_strafe\_throttle <short>)

gets the amount of strafe throttle applied by digital device stimuli

(set\_digital\_strafe\_throttle <short> <real>)

sets the amount of strafe throttle applied by digital device stimuli

(get\_digital\_yaw\_increment <short>)



gets the increment in yaw applied by digital device stimuli

(set\_digital\_yaw\_increment <short> <real>)

sets the increment in yaw applied by digital device stimuli

(get\_digital\_pitch\_increment <short>)

gets the increment in pitch applied by digital device stimuli

(set\_digital\_pitch\_increment <short> <real>)

sets the increment in pitch applied by digital device stimuli

(get\_mouse\_forward\_threshold <short>)

gets the threshold beyond which mouse movement is full forward throttle

(set\_mouse\_forward\_threshold <short> <real>)

sets the threshold beyond which mouse movement is full forward throttle

(get\_mouse\_strafe\_threshold <short>)

gets the threshold beyond which mouse movement is full strafe throttle

(set\_mouse\_strafe\_threshold <short> <real>)

sets the threshold beyond which mouse movement is full strafe throttle

(get\_mouse\_yaw\_scale <short>)

gets the scale for mouse control of yaw

(set\_mouse\_yaw\_scale <short> <real>)

sets the scale for mouse control of yaw

(get\_mouse\_pitch\_scale <short>)

gets the scale for mouse control of pitch

(set\_mouse\_pitch\_scale <short> <real>)

sets the scale for mouse control of pitch

(get\_gamepad\_forward\_threshold <short>)

gets the threshold beyond which gamepad movement is full forward throttle

(set\_gamepad\_forward\_threshold <short> <real>)

sets the threshold beyond which gamepad movement is full forward throttle

(get\_gamepad\_strafe\_threshold <short>)

gets the threshold beyond which gamepad movement is full strafe throttle

(set\_gamepad\_strafe\_threshold <short> <real>)

sets the threshold beyond which gamepad movement is full strafe throttle

(get\_gamepad\_yaw\_scale <short>)

gets the scale for gamepad control of yaw

(set\_gamepad\_yaw\_scale <short> <real>)

sets the scale for gamepad control of yaw

(get\_gamepad\_pitch\_scale <short>)

gets the scale for gamepad control of pitch

(set\_gamepad\_pitch\_scale <short> <real>)

sets the scale for gamepad control of pitch

(bind <string> <string> <string>)

binds an input device/button combination to a game control

(unbind <string> <string>)

unbinds an input device/button combination

(print\_binds)

prints a list of all input bindings

(sv\_end\_game)

End the current game.

(change\_team <short>)

change your team (0=red,1=blue,else=auto)

(sv\_mapcycle)

Print the contents of the currently loaded mapcycle file

(sv\_mapcycle\_begin)

Restart or begin playing the currently loaded mapcycle file

(sv\_mapcycle\_add <string> <string>)

Usage: sv\_mapcycle\_add <mapname> <variantname>  
Add a new game to the end of the mapcycle file.

(sv\_mapcycle\_del <long>)

Usage: sv\_mapcycle\_del <index>  
Removes the game at <index>. Will not affect running games.

(sv\_map\_next)

<Server Only> Abort the current game and begin the next game in the playlist

(sv\_map\_reset)

<Server Only> Reset the current game

(sv\_map <string> <string>)

<Server Only> Usage: "sv\_map <mapname> <variantname>"  
Abort current game and playlist and start specified game

(rcon [rcon password] [command])

Sends a command for server to execute at console. Use \" to send quotes.

(sv\_rcon\_password [remote console password])

Sets the server remote console password. If no password is given, displays the current password. Enter "" to disable rcon.

(sv\_say <string>)

<Server Only> Usage: "sv\_say <message>"  
Send a message to users

(sv\_players)

<Server Only> Print a list of players in the current game

(sv\_kick <string>)

<Server Only> Usage: sv\_kick <player # or name>  
Kicks the specified player from the server

(sv\_ban [player # or name] opt:[duration (#)(m,h,d)])

<Server Only> Player is kicked and added to banned.txt. Use sv\_players to find the index.

Specify optional duration for timed ban. Use 0 to follow sv\_ban\_penalty rules.

(sv\_banlist)

Print a list of banned players

(sv\_unban <long>)

<Server Only> Usage: sv\_unban <index>

Removes player at index in the banlist. Use sv\_banlist to find the index

(sv\_parameters\_reload)

<Server Only> Usage: sv\_parameters\_reload

Reloads the parameters.cfg file.

(sv\_parameters\_dump)

Dumps out the local parameter configuration to parameters.cfg file.

(sv\_status)

Shows status of the server

(sv\_name [name])

Sets the name of the server. If no name is given, displays the current name.

(sv\_password [password])

Sets the server password. If no password is given, displays the current password.

(sv\_log\_note <string>)

Leave a note in the server log

(sv\_log\_file [log file name])

Sets the server log file name. If no name is given, displays the current log file name.

(sv\_log\_enabled ["1" to enable, "0" to disable])

Enables or disables server logging. If 0/1 is not given, displays the current logging status.

(sv\_log\_rotation\_threshold [threshold in kilobytes])

Sets the log rotation threshold. When a log file's size (in kilobytes) exceeds this number, it will be rotated. Set to 0 to disable log rotation. If the threshold is not specified, displays the current threshold.

(sv\_log\_echo\_chat [preference])

Enables or disables chat echo to the console. Set the preference to 0 to disable chat echo, or 1 to enable chat echo. If the preference is not specified, displays the current preference.

(profile\_load <string>)

Load any included builtin profiles and create profiles on disk.

(track\_remote\_player\_position\_updates <string>)

Sets the name of the remote player whose position update are to be tracked.

(remote\_player\_stats <string>)

Displays the prediction stats of the specified remote player.

(sv\_get\_player\_action\_queue\_length <string>)

Displays the action queue length for the specified player.

(thread\_sleep <long>)

Sleeps the calling thread for the specified number of ms.

(checkpoint\_save)

save last solo checkpoint

(checkpoint\_load <string>)

load a saved checkpoint

(sv\_maplist [substring])

Display a list of maps, matching an optional substring.

(sv\_gamelist [substring])

Display a list of game types, matching an optional substring.



(sv\_friendly\_fire ["0" = defaults, "1" = off, "2" = shields, "3" = on])

Use to provide a global override for the gametype friendly fire setting.


(sv\_timelimit ["-1" = default, "0" = infinite, <time in minutes>])

Use to provide a global override for the gametype timelimit setting.

(sv\_ban\_penalty [(#)(m,h,d), 0=infinite])

Specify up to 4 ban times for repeat ban/TK offenders.

(sv\_tk\_grace [time (#)(s,m)])

Posted: Mar 6, 2006 11:30 PM  Msg. 8 of 29

(sv\_tk\_cooldown [time (#)(s,m)])

Specify a TK point cooldown period, after which players lose a TK point.

(sv\_banlist\_file [alphanumeric banlist file suffix])

Sets and opens the file to be used for the player ban list.

(sv\_maxplayers [1 - 16])

Sets the maximum number of players (between 1 and 16). If no value is given, displays the current value.

(sv\_single\_flag\_force\_reset [boolean])

Force the flag to reset in single flag CTF games when the timer expires, even if held by a player.

If not specified, displays the current value.

(sv\_motd [motd file name])

Sets the server message of the day file name. If no name is given, displays the current motd file name. Set to "" to turn motd off.

(oid\_watch)

Sets/displays translated object table watch parameters.

(oid\_dump)

dumps the whole translated object table to network.log

(oid\_status)

displays the status of the translated object table.