Tips\Notes:

Setting up your scripts:
1. To make a new script to use in a scenario, go into your data folder in your HaloCE install folder and create a new file with the extension ".hsc". Example path to a script: "data\levels\test\tutorial\scripts\test_script.hsc". Your scripts must always go into a "scripts" folder that is in your data folder that has the same path as your scenario tag in your tags folder. So the tutorial scenario path is "tags\levels\test\tutorial\tutorial.scenario", and the scripts would again go in the folder "data\levels\test\tutorial\scripts\" folder.
2. Once you have some code written in a script file open sapien and in the File menu click "Compile Scripts".
3. If there are any errors, they will appear in the game view screen.
Sapien adds all the script crap into your scenario tag so you don't have to fuck with anything in it.

Structure of a script block of code:
(script <script type> <return type> <script name>
    <code>
)

Structure of a global definition:
(global <global type> <global name> <value(s)>)

HaloScript Keywords:

Expression	Details	Usage\Example
(+ <number(s)>)	returns the sum of all specified expressions.	(+ 5 6 7 8 9) returns:35 (+ 5.5 6.6 7 8.4 9) returns:36.5
(- <number> <number>)	returns the difference of two expressions.	(- 10 5) returns:5 (- 1 0.5) returns:0.5
(* <number(s)>)	returns the product of all specified expressions.	(* 5 5) returns:25 (* 5.5 6) returns:33
(/ <number> <number>)	returns the quotient of two expressions.	(/ 10 5) returns:2 (/ 2.5 2) returns:1.25
(= <expression> <expression>)	returns true if two expressions are equal	(= (hud_get_timer_ticks) 0)
(!= <expression> <expression>)	returns true if two expressions are not equal	(!= (hud_get_timer_ticks) 0)
(> <number> <number>)	returns true if the first number is larger than the second.	(> 10 5) returns: true (> 5 10) returns: false
(< <number> <number>)	returns true if the first number is smaller than the second.	(> 4 8) returns: true (> 8 4) returns: false
(>= <number> <number>)	returns true if the first number is larger than or equal to the second.	(>= 10 10) returns: true (>= 5 10) returns: false
(<= <number> <number>)	returns true if the first number is smaller than or equal to the second.	(>= 4 4) returns: true (>= 8 4) returns: false
(and <boolean(s)>)	returns true if all specified expressions are true.	(and (player_action_test_action) true)
(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.	
(cond (<boolean1> <result1>) [(<boolean2> <result2>) [...]])	returns the value associated with the first true condition.	
(global <type> <name> <inital value>)	Makes a new global	
(if <boolean> <then> [<else>])	returns one of two values based on the value of a condition.	
(inspect <expression>)	prints the value of an expression to the screen for debugging purposes.	
(min <number(s)>)	returns the minimum of all specified expressions.	(min 1 3 2 4 5 7 6 8 9) returns: 1
(max <number(s)>)	returns the maximum of all specified expressions.	(max 1 3 2 4 5 7 6 8 9) returns:9
(not <boolean>)	returns the opposite of the expression.	
(or <boolean(s)>)	returns true if any specified expressions are true.	
(script <script type> [<return type>] <name>)	used for creating new scripts	
(set <variable name> <value>)	set the value of a global variable.	
(sleep <short> [<script>])	pauses execution of this script (or, optionally, another script) for the specified number of ticks.	(sleep 100 more_weapons)
(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.	(sleep_until false 5)
(wake <script name>)	wakes a sleeping script in the next update.	(wake more_weapons)
(thread_sleep <long>)	Sleeps the calling thread for the specified number of ms.	(thread_sleep 20) returns: sleeps for 20ms

to top

Script Types:

Script Type	Details	Example
startup	Performed only on map startup	(script startup init_shit)
dormant	Performed when something happens?	(script dormant wait_for_it)
continuous	Always being performed	(script continuous spawn_warthog)
static	Performed when called from another script	(script static multiply_5)
stub	Sort of like a prototype. Doesn't do anything. Used for overriding scripts; but overridden scripts can't have the same return type as another one.	(script stub wtf)

to top

Value Types:

Type	Details	Example
boolean	A value that is true or false	true false
real	Floating point value Value Range: 3.4E +/- 38 (6 digits)	3.000000
short	Short integer value Lowest Value: -32767 Highest Value: 32768	2

long	Long integer value Lowest Value: –2,147,483,648 Highest Value: 2,147,483,648	2000000000
string	String of characters in double quotes Max number of characters: 32?	"This is a string"
trigger_volume	A "Trigger Volumes" value (a block in the scenario tag)	
cutscene_flag	A "Cutscene Flags" value (a block in the scenario tag)	
cutscene_camera_point	A "Cutscene Camera Points" value (a block in the scenario tag)	
cutscene_title	A "Cutscene Titles" value (a block in the scenario tag)	
cutscene_recording	A "Cutscene Recording" value that isn't in the public HaloCE scenario tag?	
device_group	A "Device Groups" value (a block in the scenario tag)	
ai	A "Encounters" value? (a block in the scenario tag)	
ai_command_list	A "Command Lists" value (a block in the scenario tag)	
starting_profile	A "Player Starting Profile" value (a block in the scenario tag)	
conversation	A "AI Conversations" value (a block in the scenario tag)	
navpoint		
hud_message	Hud_message tag [hmt]	
object_list	A object list	
sound	Sound tag [snd!]	
effect	Effect tag [effe]	
damage	Damage tag [jpt!]	
looping_sound	Looping_sound tag [sndl]	
animation_graph	Animation_graph tag [antr]	
actor_variant	Actor_variant tag [actv]	
damage_effect	Damage_effect tag	
object_definition		
game_difficulty	easy normal hard impossible	
team	player human covenant flood sentinal	
ai_default_state	none sleeping alert moving guarding searching fleeing	
actor_type	ie, elite	
hud_corner	top_left top_right bottom_left bottom_right	
object	Object tag [obj]	

unit	Unit tag [unit]	
vehicle	Vehicle tag [vech]	
weapon	Weapon tag [weap]	
device	Device tag [devc]	
scenery	Scenery tag [scen]	
object_name	A "Object Names" value (a block in the scenario tag)	
unit_name	A "Bipeds" value (a block in the scenario tag)	
vehicle_name	A "Vehicles" value (a block in the scenario tag)	
weapon_name	A "Weapons" value (a block in the scenario tag)	
device_name	A "Device" value (a block in the scenario tag)	
scenery_name	A "Scenery" value (a block in the scenario tag)	

to top

Functions:

A:

Expression	Details	Usage\Example
(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	
(ai_actors <ai>)	converts an ai reference to an object list.	
(ai_allegiance <team> <team>)	creates an allegiance between two teams.	
(ai_allegiance_broken <team> <team>)	returns whether two teams have an allegiance that is currently broken by traitorous behavior	
(ai_allegiance_remove <team> <team>)	destroys an allegiance between two teams.	
(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_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_attack <ai>)	makes the specified platoon(s) go into the attacking state.	
(ai_automatic_migration_target <ai> <boolean>)	enables or disables a squad as being an automatic migration target	
(ai_berserk <ai> <boolean>)	forces a group of actors to start or stop berserking	
(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_command_list <ai> <ai_command_list>)	tells a group of actors 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_by_unit <unit> <ai_command_list>)	tells a named unit to begin executing the specified command list	
(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_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_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_conversation_stop <conversation>)	stops a conversation from playing or trying to play	
(ai_debug_communication_focus <string(s)>)	focuses (or stops focusing) a set of unit vocalization types.	
(ai_debug_communication_ignore <string(s)>)	ignores (or stops ignoring) a set of AI communication types when printing out communications.	
(ai_debug_communication_suppress <string(s)>)	suppresses (or stops suppressing) a set of AI communication types.	
(ai_debug_sound_point_set)	drops the AI debugging sound point at the camera location	(ai_debug_sound_point_set)

(ai_debug_speak <string>)	makes the currently selected AI speak a vocalization (e.g. ai_speak "pain minor")	(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")	(ai_speak_list "involuntary")
(ai_debug_teleport_to <ai>)	teleports all players to the specified encounter	
(ai_debug_vocalize <string> <string>)	makes the selected AI vocalize	
(ai_defend <ai>)	makes the specified platoon(s) go into the defending state.	
(ai_deselect)	clears the selected encounter.	(ai_deselect)
(ai_detach <unit>)	detaches the specified unit from all AI.	
(ai_dialogue_triggers <boolean>)	turns impromptu dialogue on or off.	(ai_dialogue_triggers true) (ai_dialogue_triggers false)
(ai_disregard <object_list> <boolean>)	if TRUE, forces all actors to completely disregard the specified units, otherwise lets them acknowledge the units again	(ai_disregard (players) true) (ai_disregard (players) false)
(ai_erase <ai>)	erases the specified encounter and/or squad.	
(ai_erase_all)	erases all AI.	
(ai_exit_vehicle <ai>)	tells a group of actors to get out of any vehicles that they are in	
(ai_follow_distance <ai> <real>)	sets the distance threshold which will cause squads to migrate when following someone	
(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_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_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_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_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_grenades <boolean>)	turns grenade inventory on or off.	
(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_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_lines)	cycles through AI line-spray modes	
(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_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_look_at_object <unit> <object>)	tells an actor to look at an object until further notice	
(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_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_nonswarm_count <ai>)	return the number of non-swarm actors in the specified encounter and/or squad.	
(ai_place <ai>)	places the specified encounter on the map.	
(ai_playfight <ai> <boolean>)	sets an encounter to be playfighting or not	
(ai_prefer_target <object_list> <boolean>)	if TRUE, *ALL* enemies will prefer to attack the specified units. if FALSE, removes the preference.	
(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_renew <ai>)	refreshes the health and grenade count of a group of actors, so they are as good as new	

(ai_retreat <ai>)	makes all squads in the specified platoon(s) maneuver to their designated maneuver squads.	
(ai_select <ai>)	selects the specified encounter.	
(ai_set_blind <ai> <boolean>)	enables or disables sight for actors in the specified encounter.	
(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_set_deaf <ai> <boolean>)	enables or disables hearing for actors in the specified encounter.	
(ai_set_respawn <ai> <boolean>)	enables or disables respawning in the specified encounter.	
(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_team <ai> <team>)	makes an encounter change to a new team	
(ai_spawn_actor <ai>)	spawns a single actor in the specified encounter and/or squad.	
(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_stop_looking <unit>)	tells an actor to stop looking at whatever it's looking at	
(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_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_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_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_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_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	
(attract_mode_start)	N/A in pc	

to top

B:

(bind <string> <string> <string>)	binds an input device/button combination to a game control	
(breakable_surfaces_enable <boolean>)	enables or disables breakability of all breakable surfaces on level	(breakable_surfaces_enable false)
(breakable_surfaces_enable <boolean>)	enables or disables breakability of all breakable surfaces on level	(breakable_surfaces_enable false)
(breakable_surfaces_reset)	restores all breakable surfaces	(breakable_surfaces_reset)

to top

C:

(camera_control <boolean>)	toggles script control of the camera.	(camera_control true) (camera_control false)
(camera_set <cutscene_camera_point> <short>)	moves the camera to the specified camera point over the specified number of ticks.	(camera_set somewhere_point 100)
(camera_set_animation <animation_graph> <string>)	begins a prerecorded camera animation.	
(camera_set_dead <unit>)	makes the scripted camera zoom out around a unit as if it were dead.	(camera_set_dead player0)
(camera_set_first_person <unit>)	makes the scripted camera follow a unit.	(camera_set_first_person player0)
(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_relative somewhere_point 200 warthog_mp_1
(camera_time)	returns the number of ticks remaining in the current camera interpolation.	(camera_time)
(change_team <short>)	change your team (0=red,1=blue,else=auto)	(change_team 0) returns: changes you to red (change_team 1) returns: changes you to blue (change_team 2) returns: auto balance
(cheat_active_camouflage)	gives the player active camouflage	(cheat_active_camouflage)
(cheat_active_camouflage_local_player <short>)	gives the player active camouflage	(cheat_active_camouflage_local_player 1)
(cheat_all_powerups)	drops all powerups near player	(cheat_all_powerups)
(cheat_all_vehicles)	drops all vehicles on player	(cheat_all_vehicles)
(cheat_all_weapons)	drops all weapons near player	(cheat_all_weapons)
(cheat_spawn_warthog)	drops a warthog near player	(cheat_spawn_warthog)
(cheat_teleport_to_camera)	teleports player to camera location	(cheat_teleport_to_camera)
(cheats_load)	reloads the cheats.txt file	(cheats_load)
(checkpoint_load <string>)	load a saved checkpoint	
(checkpoint_save)	save last solo checkpoint	(checkpoint_save)
(cinematic_abort)	aborts a cinematic	(cinematic_abort)

(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_start <boolean>)	starts screen effect; pass TRUE to clear	
(cinematic_screen_effect_stop)	returns control of the screen effects to the rest of the game	
(cinematic_set_near_clip_distance <real>)		
(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_show_letterbox <boolean>)	sets or removes the letterbox bars	(cinematic_show_letterbox true) (cinematic_show_letterbox false)
(cinematic_skip_start_internal)		(cinematic_skip_start_internal)
(cinematic_skip_stop_internal)		(cinematic_skip_stop_internal)
(cinematic_start)	initializes game to start a cinematic (interruptive) cutscene	(cinematic_start)
(cinematic_stop)	initializes the game to end a cinematic (interruptive) cutscene	(cinematic_stop)
(cinematic_suppress_bsp_object_creation <boolean>)	suppresses or enables the automatic creation of objects during cutscenes due to a bsp switch	(cinematic_suppress_bsp_object_creation true) (cinematic_suppress_bsp_object_creation false)
(config_one_control <string>)	test function to configure a single control	
(connect <string> <string>)	Attempt to connect to server - use ip:port password as parameters	
(crash <string>)	crashes (for debugging).	(crash "Something is wrong")
(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)	

to top

D:

(damage_new <damage> <cutscene_flag>)	causes the specified damage at the specified flag.	(damage_new "scenery\emitters\burning_flame\flame" enter_lava_flag)
(damage_object <damage> <object>)	causes the specified damage at the specified object.	(damage_object "weapons\assault rifle\bullet" player0)
(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	
(debug_camera_load)	loads the saved camera position and facing.	(debug_camera_load)
(debug_camera_save)	saves the camera position and facing.	(debug_camera_save)
(debug_memory)	dumps memory leaks.	(debug_memory)
(debug_memory_by_file)	dumps memory leaks by source file.	(debug_memory_by_file)
(debug_memory_for_file <string>)	dumps memory leaks from the specified source file.	(debug_memory_for_file "\halopc\haloce\source\tag_files\tag_groups.c")
(debug_teleport_player <short> <short>)		
(debug_sounds_distances <string> <real> <real>)	changes the minimum and maximum distances for all sound classes matching the substring.	
(debug_sounds_enable <string> <boolean>)	enables or disabled all sound classes matching the substring.	
(debug_sounds_wet <string> <real>)	changes the reverb level for all sound classes matching the substring.	
(debug_tags)	writes all memory being used by tag files into tag_dump.txt	(debug_tags)
(delete_save_game_files)	delete all custom profile files	(delete_save_game_files)
(device_get_position <device>)	gets the current position of the given device (used for devices without explicit device groups)	
(device_get_power <device>)	gets the current power of a named device	
(device_group_change_only_once_more_set <device_group> <boolean>)	TRUE allows a device to change states only once	
(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_get_position <device>)	gets the current position of the given device (used for devices without explicit device groups)	
(device_set_position <device> <real>)	set the desired position of the given device (used for devices without explicit device groups)	(device_set_position <device> 1.0)
(device_set_position_immediate <device> <real>)	instantaneously changes the position of the given device (used for devices without explicit device groups	(device_set_position_immediate <device> 1.0)
(device_set_power <device> <real>)	immediately sets the power of a named device to the given value	(device_set_power <device> 1.0)
(display_scenario_help <short>)	display in-game help dialog	(display_scenario_help 1)

to top

E:

(effect_new <effect> <cutscene_flag>)	starts the specified effect at the specified flag.	(effect_new "effects\coop teleport" teleporting_flag)
(effect_new_on_object_marker <effect> <object> <string>)	starts the specified effect on the specified object at the specified marker.	(effect_new_on_object_marker "effects\burning large" warthog_mp "driver")
(enable_hud_help_flash <boolean>)	starts/stops the help text flashing	(enable_hud_help_flash true) (enable_hud_help_flash false)
(error_overflow_suppression <boolean>)	enables or disables the suppression of error spamming	(error_overflow_suppression true) (error_overflow_suppression false)

to top

F:

(fade_in <real> <real> <real> <short>)	does a screen fade in from a particular color in the amount of ticks	(fade_in 0.0 0.0 0.0 100)
(fade_out <real> <real> <real> <short>)	does a screen fade out to a particular color in the amount of ticks	(fade_out 1.0 1.0 1.0 100)
(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	

to top

G:

(game_all_quiet)	returns FALSE if there are bad guys around, projectiles in the air, etc.	(game_all_quiet)
(game_difficulty_get)	returns the current difficulty setting, but lies to you and will never return easy, instead returning normal	(game_difficulty_get)
(game_difficulty_get_real)	returns the actual current difficulty setting without lying	(game_difficulty_get_real)
(game_difficulty_set <game_difficulty>)	changes the difficulty setting for the next map to be loaded.	(game_difficulty_set easy) (game_difficulty_set normal) (game_difficulty_set hard) (game_difficulty_set impossible)
(game_is_cooperative)	returns TRUE if the game is cooperative	(game_is_cooperative)
(game_lost)	causes the player to revert to his previous saved game	(game_lost)
(game_revert)	reverts to last saved game, if any (for testing, the first bastard that does this to me gets it in the head)	(game_revert)
(game_reverted)	don't use this for anything, you black-hearted bastards.	(game_reverted)
(game_safe_to_save)	returns FALSE if it would be a bad idea to save the player's game right now	(game_safe_to_save)
(game_safe_to_speak)	returns FALSE if it would be a bad idea to save the player's game right now	(game_safe_to_speak)
(game_save)	checks to see if it is safe to save game, then saves (gives up after 8 seconds)	(game_save)
(game_save_cancel)	cancels any pending game_save, timeout or not	(game_save_cancel)
(game_save_no_timeout)	checks to see if it is safe to save game, then saves (this version never gives up)	(game_save_no_timeout)
(game_save_totally_unsafe)	disregards player's current situation	(game_save_totally_unsafe)
(game_saving)	checks to see if the game is trying to save the map.	(game_saving)
(game_skip_ticks <short>)	skips <short> amount of game ticks. ONLY USE IN CUTSCENES!!!	(game_skip_ticks 5)

(game_speed <real>)	changes the game speed.	(game_speed 0.5)
(game_time)	gets ticks elapsed since the start of the game.	(game_time)
(game_won)	causes the player to successfully finish the current level and move to the next	(game_won)
(garbage_collect_now)	causes all garbage objects except those visible to a player to be collected immediately	(garbage_collect_now)
(get_digital_forward_throttle <short>)	gets 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	
(get_digital_yaw_increment <short>)	gets the increment in yaw applied by digital device stimuli	
(get_digital_pitch_increment <short>)	gets the increment in pitch applied by digital device stimuli	
(get_gamepad_forward_threshold <short>)	gets 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	
(get_gamepad_yaw_scale <short>)	gets the scale for gamepad control of yaw	
(get_gamepad_pitch_scale <short>)	gets the scale for gamepad control of pitch	
(get_mouse_forward_threshold <short>)	gets 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	
(get_mouse_yaw_scale <short>)	gets the scale for mouse control of yaw	
(get_mouse_pitch_scale <short>)	gets the scale for mouse control of pitch	
(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	

to top

H:

(hammer_begin <string> <string> <long> <short> <short>)	hammers the server by connecting and disconnecting repeatedly.	
(hammer_stop)	stops hammering the server.	(hammer_stop)
(help <string>)	prints a description of the named function.	(help cheats_load)
(hud_blink_health <boolean>)	starts/stops manual blinking of the health panel	(hud_blink_health true) (hud_blink_health false)
(hud_blink_motion_sensor <boolean>)	starts/stops manual blinking of the motion sensor panel	(hud_blink_motion_sensor true) (hud_blink_motion_sensor false)
(hud_blink_shield <boolean>)	starts/stops manual blinking of the shield panel	(hud_blink_shield true) (hud_blink_shield false)
(hud_clear_messages)	clears all non-state messages on the hud	(hud_clear_messages)
(hud_get_timer_ticks)	returns the ticks left on the hud timer	(hud_get_timer_ticks)
(hud_help_flash_restart)	resets the timer for the help text flashing	(hud_help_flash_restart)
(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_position <short> <short> <hud_corner>)	sets the timer upper left position to (x, y)=>(<short>, <short>)	
(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_show_crosshair <boolean>)	hides/shows the weapon crosshair	(hud_show_crosshair true) (hud_show_crosshair false)
(hud_show_health <boolean>)	hides/shows the health panel	(hud_show_health true) (hud_show_health false)
(hud_show_motion_sensor <boolean>)	hides/shows the motion sensor panel	(hud_show_motion_sensor true) (hud_show_motion_sensor false)
(hud_show_shield <boolean>)	hides/shows the shield panel	(hud_show_shield true) (hud_show_shield false)

to top

I:

(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_default <string>)	test function that looks up a default profile for a deviceid	
(input_find_joystick <string>)	test function to find a joystick by GUID (string representation)	
(input_get_joy_count)	test function to return the number of joysticks enumerated	(input_get_joy_count)
(input_is_joy_active <short>)	test function to determine if an enumerated joystick is activated or not	
(input_show_joystick_info)	test function to show the enumerated joystick information for all joystick	(input_show_joystick_info)

to top

L:

(list_count <object_list>)	returns the number of objects in a list	(list_count the_warthogs)
(list_get <object_list> <short>)	returns an item in an object list.	(list_get the_warthogs 3)

to top

M:

(magic_melee_attack)	causes player's unit to start a melee attack	(magic_melee_attack)
(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')	
(map_name <string>)	changes the name of the solo player map.	(map_name "a10")
(map_reset)	starts the map from the beginning.	(map_reset)
(message_metrics_clear)	clears network messaging metrics	(message_metrics_clear)
(message_metrics_dump <string>)	dumps network messaging metrics to given file ("" for default)	(message_metrics_dump "")
(multiplayer_map_name <string>)	changes the name of the multiplayer map	(multiplayer_map_name "schwinnzno1_alpha01a")
(net_graph_clear)	Clears the net_graph.	(net_graph_clear)

(net_graph_show <string> <string>)	Changes the net_graph display (bytes/packets, sent/received)	
(network_client_dump)	Dumps info on network client.	(network_client_dump)
(network_server_dump)	Dumps info on network server.	(network_server_dump)

to top

N:

(numeric_countdown_timer_get <short>)	<digit_index>	(numeric_countdown_timer_get 1) (numeric_countdown_timer_get -1)
(numeric_countdown_timer_restart)	Reset the timer	(numeric_countdown_timer_restart)
(numeric_countdown_timer_set <long> <boolean>)	<milliseconds>, <auto_start>	(numeric_countdown_timer_set 15500 false) (numeric_countdown_timer_set 10000 false)
(numeric_countdown_timer_stop)	Stop the timer	(numeric_countdown_timer_stop)

to top

O:

(object_beautify <object> <boolean>)	makes an object pretty for the remainder of the levels' cutscenes.	(object_beautify chief true) (object_beautify chief false)
(object_can_take_damage <object_list>)	allows an object to take damage again	(object_can_take_damage (players))
(object_cannot_take_damage <object_list>)	prevents an object from taking damage	(object_cannot_take_damage (players))
(object_create <object_name>)	creates an object from the scenario.	(object_create warthog_mp_1)
(object_create_anew <object_name>)	creates an object, destroying it first if it already exists.	(object_create_anew banshee_mp_1)
(object_create_anew_containing <string>)	creates anew all objects from the scenario whose names contain the given substring.	(object_create_anew_containing "pelican")
(object_create_containing <string>)	creates all objects from the scenario whose names contain the given substring.	(object_create_containing "warthog")
(object_destroy <object>)	destroys an object.	
(object_destroy_all)	destroys all non player objects.	(object_destroy_all)
(object_destroy_containing <string>)	destroys all objects from the scenario whose names contain the given substring.	(object_destroy_containing "pelican")
(object_pvs_activate <object>)	just another (old) name for object_pvs_set_object.	
(object_pvs_clear)	removes the special place that activates everything it sees.	(object_pvs_clear)
(object_pvs_set_camera <cutscene_camera_point>)	sets the specified cutscene camera point as the special place that activates everything it sees.	
(object_pvs_set_object <object>)	sets the specified object as the special place that activates everything it sees.	
(object_set_collideable <object> <boolean>)	FALSE prevents any object from colliding with the given object	(object_set_collideable player0 true) (object_set_collideable player0 false)
(object_set_facing <object> <cutscene_flag>)	turns the specified object in the direction of the specified flag.	(object_set_facing player0 blue_base_flag)
(object_set_melee_attack_inhibited <object> <boolean>)	FALSE prevents object from using melee attack	(object_set_melee_attack_inhibited player0 true) (object_set_melee_attack_inhibited player0 false)

(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_set_permutation player0 "right arm" ~damaged)
(object_set_ranged_attack_inhibited <object> <boolean>)	FALSE prevents object from using ranged attack	(object_set_ranged_attack_inhibited player0 true) (object_set_ranged_attack_inhibited player0 false)
(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	(object_set_scale player0 1.5 10)
(object_set_shield <object> <real>)	sets the shield vitality of the specified object (between 0 and 1).	(object_set_shield player0 1.0) ;; Set players shield to full
(object_teleport <object> <cutscene_flag>)	moves the specified object to the specified flag.	(object_teleport player0 red_base_flag)
(object_type_predict <object_definition>)	loads textures necessary to draw an object that's about to come on-screen.	
(objects_attach <object> <string> <object> <string>)	attaches the second object to the first; both strings can be empty	(objects_attach chief "right hand" ar1 "")
(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_can_see_flag the_warthogs tunnel_flag 45)
(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_object the_warthogs player0 90)
(objects_delete_by_definition <object_definition>)	deletes all objects of type <definition>	
(objects_detach <object> <object>)	detaches from the given parent object the given child object	(objects_detach chief ar1)
(objects_predict <object_list>)	loads textures necessary to draw a objects that are about to come on-screen.	(objects_predict the_bipeds)

to top

P:

(pause_hud_timer <boolean>)	pauses or unpauses the hud timer	(pause_hud_timer true) (pause_hud_timer false)
(play_update_history <long> <boolean>)	Playback client input history starting from the specified last completed update id.	
(playback)	starts game in film playback mode	(playback)
(player0_joystick_set_is_normal)	returns TRUE if player0 is using the normal joystick set	(player0_joystick_set_is_normal)
(player0_look_invert_pitch <boolean>)	invert player0's look	
(player0_look_pitch_is_inverted)	returns TRUE if player0's look pitch is inverted	(player0_look_pitch_is_inverted)
(player_action_test_accept)	returns true if any player has hit accept since the last call to (player_action_test_reset).	(player_action_test_accept)
(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_action)
(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_back)

(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_grenade_trigger)
(player_action_test_jump)	returns true if any player has jumped since the last call to (player_action_test_reset).	(player_action_test_jump)
(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_look_relative_all_directions)
(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_down)
(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_left)
(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_right)
(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_up)
(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_action_test_move_relative_all_directions)
(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_primary_trigger)
(player_action_test_reset)	resets the player action test state so that all tests will return false.	(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_zoom)
(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.	
(player_camera_control <boolean>)	enables/disables camera control globally	(player_camera_control true) (player_camera_control false)
(player_effect_set_max_rotation <real> <real> <real>)	<yaw> <pitch> <roll>	
(player_effect_set_max_translation <real> <real> <real>)	<x> <y> <z>	
(player_effect_set_max_vibrate <real> <real>)	<left> <right>	
(player_effect_start <real> <real>)	<max_intensity> <attack time>	
(player_effect_stop <real>)	<decay>	
(player_enable_input <boolean>)	toggle player input. the player can still free-look, but nothing else.	(player_enable_input true) (player_enable_input false)
(players_unzoom_all)	resets zoom levels on all players	(players_unzoom_all)
(print <string>)	prints a string to the console.	(print "50 dollars for this?!") returns: "50 dollars for this?!" to the console
(print_binds)	prints a list of all input bindings	(print_binds)

(profile_activate <string>)	activates profile sections based on a substring.	
(profile_deactivate <string>)	deactivates profile sections based on a substring.	
(profile_dump <string>)	dumps profile based on a substring.	
(profile_graph_toggle <string>)	enables or disables profile graph display of a particular value.	
(profile_graph_toggle <string>)	enables or disables profile graph display of a particular value.	
(profile_load <string>)	Load any included builtin profiles and create profiles on disk.	(profile_load "a hobo") returns: loads the profile "a hobo"
(profile_reset)	resets profiling data.	(profile_reset)
(profile_service_clear_timers)	clears the timers that are present in the profiling service	(profile_service_clear_timers)
(profile_service_dump_timers)	dumps the profiling service timers	(profile_service_dump_timers)
(profile_unlock_solo_levels)	unlocks all the solo player levels for player 1's profile	(profile_unlock_solo_levels)

to top

Q:

(quit)	quits the game	(quit)

to top

R:

(rasterizer_decals_flush)	flush all decals	(rasterizer_decals_flush)
(rasterizer_fixed_function_ambient <long>)	set the ambient light value for fixed function	(rasterizer_fixed_function_ambient 200)
(rasterizer_fps_accumulate)	average fps	(rasterizer_fps_accumulate)
(rasterizer_lights_reset_for_new_map)		
(rasterizer_model_ambient_reflection_tint <real> <real> <real> <real>)		
(rasterizer_reload_effects)	check for shader changes	(rasterizer_reload_effects)
(recording_kill <unit>)	kill the specified unit's cutscene recording.	(recording_kill player0)
(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_time <unit>)	return the time remaining in the specified unit's cutscene recording.	(recording_time player0)
(render_effects <boolean>)	Render game effects if TRUE	(render_effects true) (render_effects false)
(render_lights <boolean>)	enables/disables dynamic lights	(render_lights true) (render_lights false)

to top

S:

(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	
(script_recompile)	recompiles scripts.	(script_recompile)
(script_screen_effect_set_value <short> <real>)	sets a screen effect script value	
(set_digital_forward_throttle <short> <real>)	sets the amount of forward throttle applied by digital device stimuli	
(set_digital_pitch_increment <short> <real>)	sets the increment in pitch applied by digital device stimuli	
(set_digital_strafe_throttle <short> <real>)	sets the amount of strafe throttle applied by digital device stimuli	
(set_digital_yaw_increment <short> <real>)	sets the increment in yaw applied by digital device stimuli	
(set_gamepad_forward_threshold <short> <real>)	sets the threshold beyond which gamepad movement is full forward throttle	
(set_gamepad_pitch_scale <short> <real>)	sets the scale for gamepad control of pitch	
(set_gamepad_strafe_threshold <short> <real>)	sets the threshold beyond which gamepad movement is full strafe throttle	
(set_gamepad_yaw_scale <short> <real>)	sets the scale for gamepad control of yaw	
(set_gamma <long>)	set the gamma	(set_gamma 200)
(set_mouse_forward_threshold <short> <real>)	sets the threshold beyond which mouse movement is full forward throttle	
(set_mouse_pitch_scale <short> <real>)	sets the scale for mouse control of pitch	
(set_mouse_strafe_threshold <short> <real>)	sets the threshold beyond which mouse movement is full strafe throttle	
(set_mouse_yaw_scale <short> <real>)	sets the scale for mouse control of yaw	
(set_pitch_rate <short> <real>)	sets the yaw rate for the given player number	
(set_yaw_rate <short> <real>)	sets the yaw rate for the given player number	
(show_hud <boolean>)	shows or hides the hud	(show_hud true) (show_hud false)
(show_hud_help_text <boolean>)	shows or hides the hud help text	(show_hud_help_text true) (show_hud_help_text false)
(show_hud_timer <boolean>)	displays the hud timer	(show_hud_timer true) (show_hud_timer false)
(show_player_update_stats)	Shows update history playback stats.	(show_player_update_stats)
(sound_cache_dump_to_file)	dump dat shit!	(sound_cache_dump_to_file)
(sound_cache_flush)	i'm a rebel!	(sound_cache_flush)
(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.	
(sound_eax_enabled)	Returns true if EAX extensions are enabled	(sound_eax_enabled)
(sound_enable <boolean>)	enables or disables all sound.	(sound_enable true) (sound_enable false)
(sound_enable_eax <boolean>)	Enable or disable EAX extensions	(sound_enable_eax true) (sound_enable_eax false)
(sound_enable_hardware <boolean> <boolean>)	Enable or disable hardware sound buffers	
(sound_get_effects_gain)	Returns the game's effects gain	(sound_get_effects_gain)

(sound_get_gain <string>)	absolutely do not use this either	
(sound_get_master_gain)	Returns the game's master gain	(sound_get_master_gain)
(sound_get_music_gain)	Returns the game's music gain	(sound_get_music_gain)
(sound_get_supplementary_buffers)	Get the amount of supplementary buffers	(sound_get_supplementary_buffers)
(sound_impulse_predict <sound> <boolean>)	loads an impulse sound into the sound cache ready for playback.	(sound_impulse_predict "sound\sfx\impulse\ting\ting" true) (sound_impulse_predict "sound\sfx\impulse\ting\ting" false)
(sound_impulse_start <sound> <object> <real>)	plays an impulse sound from the specified source object (or "none"), with the specified scale.	
(sound_impulse_stop <sound>)	stops the specified impulse sound.	
(sound_impulse_time <sound>)	returns the time remaining for the specified impulse sound.	
(sound_looping_predict <looping_sound>)	your mom.	
(sound_looping_set_alternate <looping_sound> <boolean>)	enables or disables the alternate loop/alternate end for a 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_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_set_effects_gain <real>)	Set the game's effects gain	(sound_set_effects_gain 2.0)
(sound_set_env <short>)	Change environment preset	(sound_set_env 1)
(sound_set_factor <real>)	Set the DSound factor value	
(sound_set_gain <string> <real>)	absolutely do not use this	
(sound_set_master_gain <real>)	Set the game's master gain	(sound_set_master_gain 0.5)
(sound_set_music_gain <real>)	Set the game's music gain	(sound_set_music_gain 3.0)
(sound_set_rolloff <real>)	Set the DSound rolloff value	
(sound_set_supplementary_buffers <short> <boolean>)	Set the amount of supplementary buffers	
(structure_bsp_index)	returns the current structure bsp index	(structure_bsp_index)
(structure_lens_flares_place)	places lens flares in the structure bsp	(structure_lens_flares_place)
(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_banlist)
(sv_end_game)	End the current game.	(sv_end_game)
(sv_kick <string>)	<Server Only> Usage: sv_kick <player # or name> Kicks the specified player from the server	(sv_kick "Micro$oft") returns: kicks Micro$oft
(sv_map <string> <string>)	<Server Only> Usage: "sv_map <mapname> <variantname>" Abort current game and playlist and start specified game	
(sv_map_next)	<Server Only> Abort the current game and begin the next game in the playlist	(sv_map_next)
(sv_map_reset)	<Server Only> Reset the current game	(sv_map_reset)
(sv_mapcycle)	Print the contents of the currently loaded mapcycle file	(sv_mapcycle)

(sv_mapcycle_add <string> <string>)	Usage: sv_mapcycle_add <mapname> <variantname> Add a new game to the end of the mapcycle file.	
(sv_mapcycle_begin)	Restart or begin playing the currently loaded mapcycle file	(sv_mapcycle_begin)
(sv_mapcycle_del <long>)	Usage: sv_mapcycle_del <index> Removes the game at <index>. Will not affect running games.	
(sv_maxplayers [1 - 16])	Sets the maximum number of players (between 1 and 16). If no value is given, displays the current value.	(sv_maxplayers 10)
(sv_name [name])	Sets the name of the server. If no name is given, displays the current name.	(sv_name) returns: current sever name (sv_name "yousuck") returns: "yousuck" as sever name
(sv_parameters_dump)	Dumps out the local parameter configuration to parameters.cfg file.	(sv_parameters_dump)
(sv_parameters_reload)	<Server Only> Usage: sv_parameters_reload Reloads the parameters.cfg file.	(sv_parameters_reload)
(sv_password [password])	Sets the server password. If no password is given, displays the current password.	(sv_password) returns: sever password (sv_password "1234") returns: "1234" as sever password
(sv_players)	<Server Only> Print a list of players in the current game	
(sv_status)	Shows status of the server	(sv_status)
(sv_unban <long>)	<Server Only> Usage: sv_unban <index> Removes player at index in the banlist. Use sv_banlist to find the index	(sv_unban 1)
(switch_bsp <short>)	takes off your condom and changes to a different structure bsp	(switch_bsp 0)

to top

T:

(texture_cache_flush)	don't make me kick your ass	(texture_cache_flush)
(time_code_reset)	resets the time code timer	(time_code_reset)
(time_code_show <boolean>)	shows the time code timer	(time_code_show true) (time_code_show false)
(time_code_start <boolean>)	starts/stops the time code timer	(time_code_start true) (time_code_start false)

to top

U:

(ui_widget_show_path <boolean>)	blah blah	(ui_widget_show_path true) (ui_widget_show_path false)
(unbind <string> <string>)	unbinds an input device/button combination	
(unit <object>)	converts an object to a unit.	(unit chief)
(unit_aim_without_turning <unit> <boolean>)	allows a unit to aim in place without turning	
(unit_can_blink <unit> <boolean>)	allows a unit to blink or not (units never blink when they are dead)	
(unit_close <unit>)	closes the hatches on a 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)	
(unit_doesnt_drop_items <object_list>)	prevents any of the given units from dropping weapons or grenades when they die	(unit_doesnt_drop_items (players))
(unit_enter_vehicle <unit> <vehicle> <string>)	puts the specified unit in the specified vehicle (in the named seat)	(unit_enter_vehicle player0 warthog_mp_2 "gunner")
(unit_exit_vehicle <unit>)	makes a unit exit its vehicle	
(unit_get_current_flashlight_state <unit>)	gets the unit's current flashlight state	(unit_get_current_flashlight_state player0)
(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_get_health <unit>)	returns the health [0,1] of the unit, returns -1 if the unit does not exists	(unit_get_health player0)
(unit_get_shield <unit>)	returns the shield [0,1] of the unit, returns -1 if the unit does not exists	(unit_get_shield player0)
(unit_get_total_grenade_count <unit>)	returns the total number of grenades for the given unit, 0 if it does not exist	(unit_get_total_grenade_count player0)
(unit_has_weapon <unit> <object_definition>)	returns TRUE if the <unit> has <object> as a weapon, FALSE otherwise	(unit_has_weapon player0 plasma_cannon)
(unit_has_weapon_readied <unit> <object_definition>)	returns TRUE if the <unit> has <object> as the primary weapon, FALSE otherwise	(unit_has_weapon_readied player0 plasma_cannon)
(unit_impervious <object_list> <boolean>)	prevents any of the given units from being knocked around or playing ping animations	(unit_impervious (players) true) (unit_impervious (players) false)
(unit_is_playing_custom_animation <unit>)	returns TRUE if the given unit is still playing a custom animation	
(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_open <unit>)	opens the hatches on the given unit	
(unit_set_current_vitality <unit> <real> <real>)	sets a unit's current body and shield vitality	
(unit_set_desired_flashlight_state <unit> <boolean>)	sets the unit's desired flashlight state	(unit_set_desired_flashlight_state player0 true) (unit_set_desired_flashlight_state player0 false)
(unit_set_emotion <unit> <short>)	sets a unit's facial expression (-1 is none, other values depend on unit)	
(unit_set_emotion_animation <unit> <string>)	sets the emotion animation to be used for the given unit	
(unit_set_enterable_by_player <unit> <boolean>)	can be used to prevent the player from entering a vehicle	(unit_set_enterable_by_player warthog_mp_3 true) (unit_set_enterable_by_player warthog_mp_3 false)
(unit_set_maximum_vitality <unit> <real> <real>)	sets a unit's maximum body and shield vitality	
(unit_set_seat <unit> <string>)	this unit will assume the named seat	(unit_set_seat player0 "driver")

(unit_solo_player_integrated_night_vision_is_active)	returns whether the night-vision mode could be activated via the flashlight button	(unit_solo_player_integrated_night_vision_is_active)
(unit_stop_custom_animation <unit>)	stops the custom animation running on the given unit.	
(unit_suspended <unit> <boolean>)	stops gravity from working on the given unit	(unit_suspended player0 true) (unit_suspended player0 false)
(units_set_current_vitality <object_list> <real> <real>)	sets a group of units' current body and shield vitality	
(units_set_desired_flashlight_state <object_list> <boolean>)	sets the units' desired flashlight state	(units_set_desired_flashlight_state (players) true) (units_set_desired_flashlight_state (players) false)
(units_set_maximum_vitality <object_list> <real> <real>)	sets a group of units' maximum body and shield vitality	

to top

V:

(vehicle_driver <unit>)	returns the driver of a vehicle	(vehicle_driver the_ghots)
(vehicle_gunner <unit>)	returns the gunner of a vehicle	(vehicle_gunner the_warthogs)
(vehicle_hover <vehicle> <boolean>)	stops the vehicle from running real physics and runs fake hovering physics instead.	(vehicle_hover "vehicles\warthog\warthog" true) (vehicle_hover "vehicles\warthog\warthog" false)
(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_riders <unit>)	returns a list of all riders in a vehicle	(vehicle_riders the_tanks)
(vehicle_test_seat <vehicle> <string> <unit>)	tests whether the named seat has a specified unit in it	(vehicle_test_seat banshee_mp_1 "driver" player0)
(vehicle_test_seat_list <vehicle> <string> <object_list>)	tests whether the named seat has an object in the object list	(vehicle_test_seat_list ghost_mp_2 "driver" (players))
(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)	
(version)	prints the build version.	(version)
(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_object trig_volume1 player0)
(volume_test_objects <trigger_volume> <object_list>)	returns true if any of the specified objects are within the specified volume.	(volume_test_objects trig_volume2 (players))
(volume_test_objects_all <trigger_volume> <object_list>)	returns true if any of the specified objects are within the specified volume.	(volume_test_objects_all trig_volume2 (players))

to top