Counter-Strike: Condition Zero
Bot Commands and Navigation Mesh Editing...
The Definitive Guide

Bot Commands

bot_add [name]
bot_add_t [name]
bot_add_ct [name]
Causes a bot (with an optional "name" contained in the "*BotProfile.db*" bot personalities and skill levels database file) to be added to the game. "*bot_add*" will add a bot to the team specified by the "*bot_join_team*" CVar. "*bot_add_t*" and "*bot_add_ct*" forces the bot to join a specific team.

bot_kill [name, all]
This command takes either the name of a bot, or the keyword "all" (causing all bots in the game to be killed).

bot_kick [name, all]
This command takes either the name of a bot, or the keyword "all" (causing all bots in the game to be kicked).

bot_knives_only
bot_pistols_only
bot_snipers_only
bot_all_weapons
These commands are shortcuts that set the "*bot_allow_**" CVars accordingly.

bot_difficulty [0 - 3]
This CVar determines the difficulty of all newly created bots. 0 = easy, 1 = normal, 2 = hard, and 3 = expert (difficulty values higher than "3" are reset to "3"). Note that existing bots in the game will retain the difficulty setting that they were originally created with.

bot_quota [minimum number of bots]
Setting this CVar to a non-zero value will cause the given number of bots to be maintained in the game. If a bot is kicked, a new bot will be added to maintain the quota. To disable the quota, set it to zero.

bot_quota_match [ratio]
Sets the ratio of the number of bots per human player (i.e. "*number_of_bots = number_of_human_players * bot_quota_match_ratio*"). As human players join or leave the server, the number of bots in the game will be adjusted accordingly. Note that this command overrides the "*bot_quota*" command.

bot_auto_vacate [0, 1]
Set to 1 to force bots to automatically leave the server to make room for human players (set to 0 to disable).

bot_prefix [string]
The given [string] will be prefixed to all subsequently added bot names. This is useful for "clan-tagging" bots.

bot_join_team [ct, t, any]
Determines which team the bots will join.

bot_join_after_player [0, 1]
If non-zero, the bots will wait to join the game until at least one human player has joined.

bot_allow_pistols
bot_allow_shotguns
bot_allow_sub_machine_guns
bot_allow_rifles
bot_allow_snipers
bot_allow_machine_guns
bot_allow_grenades
bot_allow_shield
All of the "*bot_allow*" CVars can be either 0 or 1. If zero, the bots will not buy or use the given category of weapon.

bot_allow_rogues [0, 1]
If non-zero, allows bots to occasionally "go rogue". Rogue bots just "run and gun", and will respond to all radio commands with "Negative".

bot_defer_to_human [0, 1]
If non-zero, forces the bots to defer round goals to human players (bomb carrier, etc...).

bot_walk [0, 1]
Force all bots to walk (disallow running).

bot_stop [0, 1]
If non-zero, all bots will stop moving and responding (pauses all bots).

hostage_stop [0, 1]
If non-zero, all hostages will stop moving and responding (pauses all hostages).

bot_chatter [normal, minimal, radio, off]
Sets the amount of bot radio-chat (*Normal*, *Minimal*, *Standard Radio*, or *Off*).

bot_profile_db [filename.db]
Sets the filename of the database that defines the bot personalities and skill levels. Note that the default database filename is "BotProfile.db".

bot_about
Displays the bot module version number, and information about the bot's author.

Navigation Mesh Editing

Each of the following "*bot_nav_**" commands operate on the navigation mesh, allowing hand-tuning of the automatically learned data. It is recommended that these commands be bound to keys for ease-of-use while editing. Note that there is no "undo" operation, so save your navigation mesh frequently!

bot_nav_edit [0, 1]
Setting this CVar to 1 allows hand-tuning of the bot's navigation mesh. Once edit mode has been activated, the "*bot_nav_**" commands can be used.

bot_nav_zdraw [height value]
This value determines how high above the ground to draw the navigation "mesh" when in navigation edit mode. If the terrain is very irregular or highly sloped, it can be useful to increase this value to 10 or 15. The default value is 4.

bot_nav_mark
Marks the currently selected navigation area for later operations.

bot_nav_warp
Warps (teleports) your view to the currently marked navigation mesh. Note that you must be in "Free Look" spectator mode for this command to function.

bot_nav_delete
Deletes the currently selected navigation area.

bot_nav_split
Splits the currently selected navigation area into two new navigation areas, along the white split line.

bot_nav_merge
Merges the currently selected navigation area and a previously marked navigation area into a new, single navigation area. The merge will only occur if the two areas are the same size along the merge line.

bot_nav_connect
Creates a "*one-way*" link from the currently marked area to the currently selected area, telling the bots they can walk "*from*" the marked area "*to*" the selected area. For most areas, you will want to connect the areas in both directions. However, for some "jump down" areas, you will want the bots to be able to move one way, but not be able to get back to the other.

bot_nav_disconnect
Disconnects all connections from the currently marked area to the currently selected area.

bot_nav_begin_area
bot_nav_end_area
These two commands allow the creation of new navigation areas. "*bot_nav_begin_area*" marks one corner of the area. "*bot_nav_end_area*" marks the opposite corner of the area and creates it. To cancel the operation, issue the "*bot_nav_begin_area*" command again.

bot_nav_splice
Creates a new navigation area between the currently marked area and the currently selected area, and bidirectionally connects the new area. This command is especially useful for creating sloped navigation areas.

bot_nav_corner_select
bot_nav_corner_raise
bot_nav_corner_lower
Selects one of four corners of a currently marked navigation mesh ("*bot_nav_mark*"), and then allow the selected corner to be raised or lowered in elevation. Issue the "*bot_nav_corner_select*" command multiple times to cycle between the four corners (and then once again to deselect all of the corners).

bot_nav_crouch
Flags the currently selected area as "crouch", requiring bots to crouch (duck) to move through it.

bot_nav_jump
bot_nav_no_jump
Flags the currently selected area as "jump", or "no jump". This is a hint to the bots that they should (or should not) jump to traverse this area.

bot_nav_precise
Flags the currently selected area as "precise", requiring bots to precisely pass through this navigation mesh before continuing to the next one. This command is especially useful for small catwalks that are easy to fall from.

bot_nav_strip
Strips all approach points, encounter spots, and hiding spots (which are generated using the "*bot_nav_analyze*" command) from the currently marked navigation mesh.

Navigation Mesh Place Naming

Note that place names are used for bot radio-chat locations, as well as location text strings for player radio-chat. Also remember to save the navigation mesh often ("*bot_nav_save*") just in case you make a mistake, and then analyze the navigation mesh again ("*bot_nav_analyze*").

bot_nav_toggle_place_mode
Switches into place naming mode. Every navigation mesh that is highlighted with the crosshairs will be displayed with one of three colors:

Green = *Currently set place name*.
Blue = *Other place name already set*.
Red = *Unset place name*.

Enter command a second time to return to navigation mesh edit mode.

bot_nav_use_place
Displays the following list of usable place names:

```
Apartment          Apartments         Atrium             Attic
Back               BackAlley          BackDoor           BackHall
BackRoom           BackWay            BackYard           Balcony
Basement           Bathroom           Bedroom            BigOffice
BombsiteA          BombsiteB          BombsiteC          Bridge
Bunker             ComputerRoom       ConferenceRoom     Courtyard
Crates             Crawlspace         CTSpawn            Deck
Den                DoubleDoors        Downstairs         Ducts
Dumpster           Elevator           Entrance           Entryway
FamilyRoom         FarSide            Fence              Foyer
Front              FrontDoor          FrontHall          FrontRoom
FrontYard          Garage             Gate               GateHouse
GuardHouse         HostageRescueZone  Hostages           House
Inside             Kitchen            Ladder             LittleOffice
LivingRoom         LoadingDock        Lobby              Loft
LongHall           MainHall           Market             MeetingRoom
Middle             Mines              Office             Outside
Overpass           Patio              Porch              ProjectorRoom
Ramp               Rear               Roof               SecurityDoors
Sewers             Side               SideAlley          SideDoor
SideHall           SideRoom           SideYard           Stairs
Stairwell          StorageRoom        Tower              Truck
TSpawn             Tunnel             Underground        Underpass
Upstairs           Vault              VendingMachines    Village
VipRescueZone      Wall               Water              Window
Windows            WineCellar
```

bot_nav_use_place [place name]
Select place name (e.g. "*bot_nav_use_place BombsiteA*").

bot_nav_toggle_place_painting
Starts painting the navigation meshes with the selected place name. Note that every navigation mesh that is highlighted with the crosshairs will turn green and be set to the selected place name. Enter command a second time to stop painting. Then you can select a new place name ("*bot_nav_use_place [place name]*"), and start naming other navigation meshes ("*bot_nav_toggle_place_painting*").

bot_nav_place_floodfill
Sets the place name of the currently highlighted navigation mesh and flood fills outward from there, setting all adjacent

navigation meshes to that place name (until it encounters other navigation meshes with different place name labels).

bot_nav_place_pick
Selects the place name of the currently highlighted navigation mesh as your current place name (similar to any paint program's color-palette "eyedropper" tool).

bot_nav_mark_unnamed
Marks the closest unnamed navigation mesh, and displays the total number of unnamed meshes. This command is useful for locating unnamed navigation meshes, when used in conjuction with the "*bot_zombie 1*" and "*bot_goto_mark*" commands (as well as the "*bot_nav_warp*" command).

Navigation Mesh Processing

bot_nav_analyze
Analyze the navigation mesh to determine approach points, encounter spots, and hiding spots. This may take several minutes based on the size and complexity of the map. Note that this command requires one bot to be in the game. The recommended procedure is to save the mesh, add a bot, and then quickly enter "*bot_nav_analyze*".

bot_nav_load
Clears the current navigation mesh, and loads it from disk.

bot_nav_save
Saves the current navigation mesh to disk. The navigation mesh (".nav" file) is automatically named to correspond to the current map file. For instance, if the map is DE_Dust.bsp, the navigation file will be DE_Dust.nav.

bot_quicksave [0, 1]
If non-zero, the analysis phase of map learning will be skipped. This is useful when iteratively hand-tuning navigation files. Note that withough this analysis, the bots will not look around the world properly.

Navigation Mesh Debugging

bot_show_nav [0, 1]
If non-zero, the navigation mesh near each bot is drawn.

bot_show_danger [0, 1]
If non-zero, the "danger" in each navigation area is drawn as a vertical line. Blue lines represent danger for the Counter-Terrorists, and red lines are danger for the Terrorists.

bot_zombie [0, 1]
Causes the bots on the map to ignore danger (and other environmental variables contained in the map), and only use geometric distance for navigating around the map. This command is useful for testing the walkability of specific portions of the navigation mesh (or locating unnamed navigation meshes when editing navigation mesh place names), when used in conjuction with "*bot_nav_mark*" (or "*bot_nav_mark_unnamed*") and the command "*bot_goto_mark*".

bot_goto_mark
Causes the bots on the map to move to the center of the currently marked area. This command is useful for testing the walkability of specific portions of the navigation mesh when used in conjunction the the "*bot_zombie 1*" command.

bot_nav_check_consistency [filename.nav]
Checks the consistency and integrity of a bot navigation mesh file.

bot_traceview [0, 1]
Used for internal debugging of the currently spectated bot's navigation. The cyan line indicates the direction that the bot is looking, the purple line indicates the bot's field of view, while the orange line indicates the next navigation mesh that the bot is heading towards (as well as the bot's complete future path). Note that you must be in "First Person" or "Free Chase Cam" spectator mode for this command to function.

bot_debug [0, 1, 2]
Used for internal debugging of bot behavior. Set to 1 to display debug messages for the currently spectated bot (note that you must be in "First Person" or "Free Chase Cam" spectator mode for this command to function), or set to 2 display debug messages for all of the bots. Debugging messages will be printed to the console displaying what the bot is currently thinking and doing, as well as any error or warning messages (including an audible warning sound) pertaining to the bot's behavior or navigation.

hostage_debug [0, 1]
Used for internal debugging of the hostage's navigation. The purple line indicates the direction that the hostage is looking, the green lines indicate the hostage's field of view, the orange line indicates the next navigation mesh that the hostage is heading towards, while the yellow line indicates the hostage's complete future path. Note that you should be playing as a Counter-Terrorist, and have already pressed the "*Use*" key on a hostage (to instruct him to follow you) for best results when using this command.

Lastly, note that this command will also display debug messages that will be printed to the console displaying what the hostage is currently thinking and doing (just like the "*bot_debug*" command above), as well as any error or warning messages pertaining to the hostage's behavior or navigation.

bot_memory_usage
Displays memory usage statistics about the active bot module and the currently loaded navigation mesh file.

Rich ¥Weeds¥ Nagel
richnagel@centurylink.net
http://www.richnagel.net
http://steamcommunity.com/id/RichNagel