**Modding Tutorial for Jv\_Bots**

To learn about scripting with regular AI**,** [go here!](http://mohdatabase.wikispaces.com/AI_general_tut)

**Table of Contents**

[Modding Tutorial for Jv\_Bots](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots)

[Introduction](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Introduction)

[Method 1 - Spawnflags](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%201%20-%20Spawnflags)

[Usage](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%201%20-%20Spawnflags-Usage)

[1. Set up the trigger](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%201%20-%20Spawnflags-Usage-1.%20Set%20up%20the%20trigger)

[2. Setting up the setthread](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%201%20-%20Spawnflags-Usage-2.%20Setting%20up%20the%20setthread)

[Commands - Can and Cannot](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%201%20-%20Spawnflags-Commands%20-%20Can%20and%20Cannot)

[Cannot](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%201%20-%20Spawnflags-Commands%20-%20Can%20and%20Cannot-Cannot)

[Can](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%201%20-%20Spawnflags-Commands%20-%20Can%20and%20Cannot-Can)

[Method 2 - Botscript Method](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%202%20-%20Botscript%20Method)

[Setup in Botscript](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%202%20-%20Botscript%20Method-Setup%20in%20Botscript)

[Setup in Mapscript](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%202%20-%20Botscript%20Method-Setup%20in%20Mapscript)

[1. Setup botscript trigger properties](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%202%20-%20Botscript%20Method-Setup%20in%20Mapscript-1.%20Setup%20botscript%20trigger%20properties)

[2. Setup setthread](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%202%20-%20Botscript%20Method-Setup%20in%20Mapscript-2.%20Setup%20setthread)

[3. Stopping attraction](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%202%20-%20Botscript%20Method-Setup%20in%20Mapscript-3.%20Stopping%20attraction)

[4. Known working commands](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%202%20-%20Botscript%20Method-Setup%20in%20Mapscript-4.%20Known%20working%20commands)

[5. Examples and Important notes](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Method%202%20-%20Botscript%20Method-Setup%20in%20Mapscript-5.%20Examples%20and%20Important%20notes)

[General Scripting Tips](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-General%20Scripting%20Tips)

[Switching Teams](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-General%20Scripting%20Tips-Switching%20Teams)

[1. Switch triggerteam](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-General%20Scripting%20Tips-Switching%20Teams-1.%20Switch%20triggerteam)

[2. Making it work](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-General%20Scripting%20Tips-Switching%20Teams-2.%20Making%20it%20work)

[Objective Campnodes](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-General%20Scripting%20Tips-Objective%20Campnodes)

[Disabling Spawnpoints](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-General%20Scripting%20Tips-Disabling%20Spawnpoints)

[Setting Botproperties](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-General%20Scripting%20Tips-Setting%20Botproperties)

[Defining bots (No Trigger)](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Defining%20bots%20%28No%20Trigger%29)

[Source](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Source)

[Download](http://mohdatabase.wikispaces.com/Bots+-+Scripting#Modding%20Tutorial%20for%20Jv_Bots-Download)

**Introduction**

There are 2 methods of making bots activate triggers, teleport them, alternative objectives...  
As well as some **extremely** useful tips!  
  
Bots are untargeted by normal triggers and commands, but by using the 2 methods below, we can define bots and use (not all) a lot of commands on them. Only a few are known at the moment.  
  
Note that each method is used for a different purpose. You may use those given methods for any purpose, but it's recommended to use them for the purposes this article defines.  
  
The first is to make a bot trigger a trigger,that makes him do a series of working commands or makes him undergo a series of working commands.  
The second makes a bot use a trigger to complete alternative objectives, such as taking a document, contacting a radio...

**Method 1 - Spawnflags**

Spawnflags 12 is a spawnflag value that gives certain properties and specifications to a trigger. Spawnflags 12 mixes the ability to detect players and detecting AI and defining certain commands.

**Usage**

If you want a bot to trigger a trigger to make him undergo/do a certain set of commands or make some things happen.

**1. Set up the trigger**

spawn trigger\_(kind) "targetname" "(name)" "spawnflags" "12"

$(name).origin = ( X Y Z )

$(name) setsize ( -X -Y -Z ) ( X Y Z )

$(name) setthread (threadname)

Fill in everything between the brackets. Including the X Y Z coordinates with the corresponding values you've chosen.  
Let's continue to the setthread.

**2. Setting up the setthread**

(threadname):

local.ent = parm.other //Defines the bot

// your actions

end

Just like you would with players, you define bots the same way. In addition, you use a different name. In this case local.ent. You can use another name if you prefer but this is the general name for defining bots. Now to see what you can do exactly with those bots, continue reading.  
  
**Commands - Can and Cannot**

A lot of commands which usually only apply to players now apply to bots.

**Cannot**

All Player class commands do not work on AI. One of the most missed is *tele* of course. Neither does gravity.

-tele ( X Y Z ) // teleports a player to a specified location

-gravity // sets the gravity of a player

But fortunately, there's a working workaround for both commands!

**Can**

-local.ent.dmteam = "(team)" //not spectator

-local.ent.origin = ( X X X ) //This does the exact same thing as tele

-local.ent thread //you can use self in the executed thread as substitute for local.ent

-local.thing attach local.ent "Bip01 Spine2" 0 //attach an object to any valid spot on the bot!

-local.thing detach local.ent //detach...no need to specify the location here

-local.ent physics\_off //freeze bot

-local.ent physics\_on //unfreeze bot

-local.ent glue local.thing //glue something to the bot or glue the bot to something

-local.ent unglue local.thing

-local.ent health X //set the bots health, X being the value

-local.thing = spawn script\_origin //Use this to set the gravity of a bot, spawn a script\_origin

local.thing.origin = local.ent.origin //glue it to the bot using this and the following command

local.ent glue local.thing

local.thing gravity .1 //set the gravity to the script\_origin

-local.thing gravity 1 //Use this to set the gravity back to normal

local.thing remove

A sum of useful ones. Basically every command for the Entity, Sentient & Actor classes works on the AI.

**Method 2 - Botscript Method**

http://i1215.photobucket.com/albums/cc502/soldierofra/icon-circle-check-green-001-1.png**Requirement**: You must have a botmap made with BdBodger's Botmapper for the second method. Recognizable by a .bot file located within the obj or dm folder of your .pk3 file.

In some cases, to make bots do an existing objective (**NOT** placing a bomb but like taking documents, contacting radio's), you must make a trigger\_(kind) in the .bot script and a triggernode targetting the trigger. Recommended a trigger\_use unless you need another kind of trigger.

**Setup in Botscript**

Example:

//<------------------------------------------------------------->

level.obj\_model[108] = spawn trigger\_use //you'd probably need a trigger\_use

level.obj\_model[108].targetname = example\_trigger //targetname

level.obj\_model[108].origin = ( -932 3159 66 ) //same as the objective

level.obj\_model[108].angles = ( 0.000 0.000 0.000 ) //leave it 0, unimportant

level.obj\_model[108] setsize ( -50 -50 0 )( 50 50 50 ) //size of the trigger

level.obj\_model[108].triggerteam = none //leave it none if you want to set it in the mapscript

level.obj\_model[108].weight = 2.000 //'attraction value'

//<------------------------------------------------------------->

level.obj\_model[109] = spawn info\_pathnode //DON'T CHANGE

level.obj\_model[109].targetname = triggernode //DON'T CHANGE

level.obj\_model[109].origin = ( -932 3159 66 ) //same as trigger

level.obj\_model[109].angles = ( 0.000 180.000 0.000 )

level.obj\_model[109].target = example\_trigger //MUST target the trigger

//<------------------------------------------------------------->

This is the basic setup of a simple alternative objective trigger. You must always need these 2 parts. The first is the trigger and the second is the pathnode targetting the trigger.  
  
**NOTE**: Make sure the values [X] don't match any other value of the level.obj\_model[] list in your .bot script!!!

**Setup in Mapscript**

**1. Setup botscript trigger properties**

If you haven't set the .triggerteam property in the botscript and left it none; In the mapscript you set the triggerteam first after level waittill spawn.

$(botscript\_trigger\_targetname).triggerteam = allies //or axis

If you want to reset the trigger weight, in case you need it to change it (multiple times) in your mapscript:

$(botscript\_trigger\_targetname).weight = 20

It is **required** to set the setthread of your trigger:

$(botscript\_trigger\_targetname) thread (yoursetthread)

**NOTE**: It is important to maintain the order of the inserted values (even if you haven't used all).

**2. Setup setthread**

Of course when using this method, there are a lot less things a bot can do because there's no spawnflags 12, and when you add it, spawnflags doesn't work(ref). But since this is only used for alternate objectives the bot doesn't need to know much commands.

(yoursetthread):

self waittill trigger

local.ent = self.other

Always start with that. Without the

self waittill trigger

this won't work. And without the

local.ent = self.other

this won't work either. They must be together.

**3. Stopping attraction**

Strangely, the triggernode who targets the trigger is what attracts the bots. However it uses the weight of the trigger itself.  
But if the weight is 0, the triggernode will still attract bots. Using a big negative value will keep the bots away.  
  
So when the objective is completed, you avoid deleting the trigger and you spare you 100kb of console spam by using this to keep the bots away.

$(botscript\_trigger\_targetname).weight = -2000

This'll stop the trigger from attracting bots, any bots who were initially headed for the trigger will go somewhere else if the bot got a camp or sniper spot or another triggernode... Sometimes some bots tend to just stand still until shot and respawned and some might move after a while.

**4. Known working commands**

-if statement

-local.ent.dmteam = "(team)" //of course not spectator

-local.ent.origin = ( X X X ) //This does the exact same thing as tele

-local.ent playsound //plays a small sound once

**5. Examples and Important notes**

Example of an "Stealing-Documents" type objective:

(yoursetthread):

self waittill trigger

local.ent = self.other

if ( local.ent.dmteam == "allies" )

{

$(botscript\_trigger\_targetname) nottriggerable //you don't want them to trigger it again after completion

$(playertrigger\_targetname) nottriggerable //you don't want any player being able to "take" it after completion

level.docs\_taken++ //You'd probably need this for a winner check

$documents hide //Hides the documents, so it'll seem it's been picked up

iprintln "The allies have taken the documents!"

thread (threadname\_ofyour\_wincheck) //You probably want to know if someone has won yet

$(botscript\_trigger\_targetname).weight = -1000 //After completion, this stops bots from gathering near the trigger.

}

end

As you can see, you must think about a few things when setting an objective for bots.  
  
**NOTE**: Since your using a trigger for the objective for bots as well as player you must disable the bots trigger in the trigger for the player. Because if a player completes the objective before a bot and you don't apply this, the bot will "complete" the objective again.  
  
Example:

(playerstrigger\_setthread):

local.player = parm.other

if ( local.player.dmteam == "allies" )

{

$(playertrigger\_targetname) nottriggerable

$(botscript\_trigger\_targetname) nottriggerable //you don't want bots completing the objective again

$(botscript\_trigger\_targetname).weight = -10000 //you don't want bots gathering there

level.docs\_taken++

$documents hide

iprintln "The allies have taken the documents!"

thread (threadname\_ofyour\_wincheck)

}

end

**General Scripting Tips**

**Switching Teams**

If your objective is a switch-objective(an objective where if in the process of being completed, the opposing team must try and undo it).  
For example, the allies have to switch the radio on to send a message, the process takes 45 seconds. During those seconds, the axis have  
the chance to stop the message from being sent. Then the allies try again etc...  
  
Then you'll have to use this way which saves you even more triggers and commands.

**1. Switch triggerteam**

Now it's very simple to do this.  
When a team has turned the radio on, where you set the variable of the radio, (eg: level.radio\_contacted = 1) place the following:

$(botscript\_trigger).triggerteam = axis //or allies

Of course fill in the original trigger name. Do the same for the other team. This is switching "attraction" if the allies have done their job they leave and the axis get attracted to stop them. If they stopped they allies, the axis leave and the allies are going to try again etc...

**2. Making it work**

Now, like this it won't work long. Because when the allies activate the trigger, and the axis are trying to deactivate it, nothing will happen.  
Because the attraction may have switched of team but the trigger isn't that's why you need to refresh the trigger's memory.  
  
There are 2 ways:  
**1.**  
Refresh it by repeating it's setthread:

$(botscript\_trigger) thread (yoursetthread)

behind the teamswitch bit. There's a known issue where the switching stops working after a while.  
The **next method** hasn't got any downsides or bugs so far; so it might be good to use that one if this one stops working or if you don't want to find out.  
  
**2.**  
Put your entire script (in this case setthread) between:

while(1)

{

//your script

wait 0.15

}

end

**Note:** Do not use end the thread in your while(1) loop! Unless this is a conscious decision. Because if you end the loop, it won't loop again unless someone triggers it again. Since the switching is then disabled, the loop might not get started anymore.  
  
**Note:** Make sure the parts of the script only get activated under a certain condition and not always! To avoid lag and bugs.  
You may also change the wait if it's too short or too long (to avoid some lag).  
  
**Don't Forget**  
Of course, you must set another if statement between the brackets besides your teamswitch objective. Because if the objective is done, we  
don't need this loop anymore.  
  
Example:

if ( level.obj\_complete == 1 )

{

$(botscript\_trigger) nottriggerable

$(human\_trigger) nottriggerable

$(botscript\_trigger).weight = -2000

end //important to stop while(1)

}

This is very similar to setting bombs. Jv's botscript was originally designed to bot stock maps (with only bomb-type objectives) so all these other objectives weren't initially supported. But this article shows you it doesn't make them impossible.

**Objective Campnodes**

Now like bomb-objective, there are bombcamps. If the bomb isn't set, the defending team takes a spot or two around the bomb to defend.  
If the bomb is set by the attacking team, they'll take those positions to make sure it stays set until explosion.  
There are possibilities to applying this to non-bomb type objectives.  
  
You might have noticed but all camper and sniper nodes have the same targetname. The following for statements, counts these, while the  
if statements, identifies their tag and applies changes. Tags are similar to nametags, they tell us these nodes (with the same tags)  
belong together in a "group" we've set. So we'll only give a certain tag to identify to the sniper/campnodes we're using for alternative objectives.

for(local.i = 1; local.i <= $camper.size; local.i++)

{

if($camper[local.i].tag == tag1 )

{

$camper[local.i].noaxis = 1

$camper[local.i].noallies = 0

}

}

for(local.i = 1; local.i <= $sniper.size; local.i++)

{

if( $sniper[local.i].tag == tag1 )

{

$sniper[local.i].noaxis = 1

$sniper[local.i].noallies = 0

}

}

You may(/must) place this in the same setthread I showed you earlier using the while(1) loop to refresh.  
  
**Note:** Make sure this only happens under a certain condition and not always! To avoid lag and bugs.  
You may also change the wait if it's too short or too long (to avoid some lag).

**Disabling Spawnpoints**

**Note:** Untested but should work; suggested by jv\_map.  
  
You can disable and enable spawnpoints for gameplay reasons, gametype reasons...  
  
First, all spawnpoints have 2 targetnames: axisspawn and alliesspawn.  
Therefor we can't just disable anything because it'll disable all. So let's use the tag trick again:  
Example of a botspawn:

level.obj\_model[402] = spawn script\_origin

level.obj\_model[402].targetname = alliesspawn

level.obj\_model[402].origin = ( X X X )

level.obj\_model[402].angles = ( 0.000 96.000 0.000 )

level.obj\_model[402].tag = disableme

Then in your script, in the part where you want to disable/enable the spawn:

for(local.i = 1; local.i <= $alliesspawn.size; local.i++)

{

if($alliesspawn[local.i].tag == disableme )

{

$alliesspawn[local.i].disablespawn = 1

}

}

To enable do the exact same thing but instead change the disablespawn value to 0. You might need to use a while loop again.

**Setting Botproperties**

Since the destination of bots is determined the second a bot spawns, you can set special properties to the node the bot is headed to so the bot'll enherit these. Spawn a campernode, snipernode, bombcamp... And use on of these known and working commands:

level.obj\_model[X].hearing = 1000

level.obj\_model[X].sight = 1000

level.obj\_model[X].gren\_awareness = 150

level.obj\_model[X].accuracy = 150

level.obj\_model[X].ammo\_grenade = 5

level.obj\_model[X].health = 120

Commands are pretty straight forward. Normally bots have unlimited grenades but using the ammo\_grenade property you can limit it. Sight/hearing uses the game units in the context of a radius. In this case the bot can see as far as 1000 game units in any direction.  
Replace the X with the number of the level.obj\_model which spawns the info\_pathnode of the camper/sniper/bombcamp node.  
  
Example:

level.obj\_model[54] = spawn info\_pathnode

level.obj\_model[54].targetname = sniper

level.obj\_model[54].origin = ( X X X )

level.obj\_model[54].angles = ( 0.000 0.000 0.000 )

level.obj\_model[54].target = so16

level.obj\_model[54].noaxis = 1

level.obj\_model[54].noallies = 0

level.obj\_model[54].weight = 1.000

level.obj\_model[54].nocrouch = 0

//set special properties

level.obj\_model[54].hearing = 1000

level.obj\_model[54].sight = 1000

There are more commands, but they are still untested.

**Defining bots (No Trigger)**

Of course all scripting is NOT done through triggers and you probably wonder how this can be done. Similar to the player's counting system; this'll count and define bots. Though it is still unknown what commands work and don't work. But there a quite a few which would need to work.  
  
Bot counting system for non-trigger use:

//if (level.jvbot\_jv\_mp\_ai\_running == 1 && level.botlastid != NIL)

//{

if(level.botlastid != NIL)

{

for(local.i = 1; local.i <= level.botlastid; local.i++)

{

local.bot = level.actualbots[local.i]

//script

}

}

//}

end

Use the commented out lines to avoid some errors in the console when jv\_bot's scripts aren't installed or activated. If you do, then comment the following in the script above out:

if (level.jvbot\_jv\_mp\_ai\_running == 1 && level.botlastid != NIL)

{

// if(level.botlastid != NIL)

// {

for(local.i = 1; local.i <= level.botlastid; local.i++)

{

local.bot = level.actualbots[local.i]

//script

}

// }

}

end

To disable and enable weapons, use the weaponpriorities bdbodger's botmapper generates and is set in all botmaps.  
To give them all weapons, that would make conflicts with the jv\_bot script itself!  
  
This method has been used before in the Admin Pro Menu to make it work with bots.

**Source**

**Source Post** in [Mods-R-Us.net -> Forums -> Scripting -> Topic: BotScripting Unwrapped](http://www.mods-r-us.net/forums.php?m=posts&q=4185&n=last#bottom) by Soldier of Ra (of [MoHAAirborne.co.cc](http://www.mohaairborne.co.cc/))  
**Thanks to:** Sor (Soldier of Ra), PROXIMO and Cobra {SFX}

Comments from $or:

That article (above) is wrong about spawnflags 12... or rather, I was wrong when I wrote the article.  
  
If I recall correctly spawnflags 12 should trigger only for non-players. Try adding spawnflags 8 instead.  
  
If that doesn't work, I recommend you use the 'damage' event. It should be raised not only for players   
and actors, but for each damageable entity:  
[http://www.x-null.net/forums/showthr...-Documentation](http://www.x-null.net/forums/showthread.php?728-New-EventSystem-Engine-Documentation)

**References** *From entdefs.pk3*

|  |  |  |
| --- | --- | --- |
| **SpawnFlag** | **Option** | **Description** |
| 1 | / |  |
| 2 | / |  |
| 4 | NOT\_PLAYERS | the trigger does not respond to players |
| 8 | MONSTERS | the trigger will respond to monsters |
| 16 | PROJECTILES | the trigger will respond to projectiles (rockets, grenades, etc.) |
| 32 | / |  |
| 64 | / |  |
| 128 | DAMAGE | the trigger will only respond to bullets (at least, that's what it says) |

*For the damage event:*

|  |  |
| --- | --- |
| **Argument** | **Description** |
| Target | the entity that is inflicting the damage on *Entity* (may or may not be a player or actor entity, it may be the weapon itself or some other entity) |
| Inflictor | the entity that caused damage to be inflicted upon *Entity* |
| Damage | the amount of damage (in health points) that is inflicted on *Entity* |
| Position | the origin to which the damage is directed (not used often) |
| Direction | the directionality of the damage (usually a normalized vector) |
| Normal | the normal vector of a plane (not sure which plane, but ignore since it is almost always (0 0 0)) |
| Knockback | the amount of force exerted in *Direction* |
| DamageFlags | how the damage is inflicted on *Entity* |
| MeansOfDeath | what kind of damage is inflicted on *Entity* |
| Location | where the damage is inflicted on *Entity* (i.e. its model) |
| Entity | the entity that takes the damage (may or may not be a player or actor entity, it may be any damageable entity) |

*From pak0.pk3/global/bullethit.scr:*

|  |  |  |
| --- | --- | --- |
| **DamageFlag** | **Option** | **Description** |
| 1 | DAMAGE\_RADIUS | damage was indirect |
| 2 | DAMAGE\_NO\_ARMOR | armour does not protect from this damage |
| 4 | DAMAGE\_ENERGY | damage is from an energy based weapon |
| 8 | DAMAGE\_NO\_KNOCKBACK | do not affect velocity, just view angles |
| 16 | DAMAGE\_BULLET | damage is from a bullet (used for ricochets) |
| 32 | DAMAGE\_NO\_PROTECTION | armor, shields, invulnerability and godmode have no effect |
| 64 | DAMAGE\_NO\_SKILL | damage is not affected by skill level |

|  |  |
| --- | --- |
| **Means of Death** | **Description** |
| 0 | MOD\_NONE |
| 1 | MOD\_SUICIDE |
| 2 | MOD\_CRUSH |
| 3 | MOD\_CRUSH\_EVERY\_FRAME |
| 4 | MOD\_TELEFRAG |
| 5 | MOD\_LAVA |
| 6 | MOD\_SLIME |
| 7 | MOD\_FALLING |
| 8 | MOD\_LAST\_SELF\_INFLICTED |
| 9 | MOD\_EXPLOSION |
| 10 | MOD\_EXPLODEWALL |
| 11 | MOD\_ELECTRIC |
| 12 | MOD\_ELECTRICWATER |
| 13 | MOD\_THROWNOBJECT |
| 14 | MOD\_BEAM |
| 15 | MOD\_ROCKET |
| 16 | MOD\_IMPACT |
| 17 | MOD\_BULLET |
| 18 | MOD\_FAST\_BULLET |
| 19 | MOD\_VEHICLE |
| 20 | MOD\_FIRE |
| 21 | MOD\_FLASHBANG |
| 22 | MOD\_ON\_FIRE |
| 23 | MOD\_GIB |
| 24 | MOD\_IMPALE |
| 25 | MOD\_BASH |
| 26 | MOD\_TOTAL\_NUMBER |

|  |  |
| --- | --- |
| **Location** | **Description** |
| -1 | Unknown |
| 0 | Pelvis |
| 1 | Lower Torso |
| 2 | Mid Torso |
| 3 | Upper Torso |
| 4 | Neck |
| 5 | Head |
| 6 | Right Upper Arm |
| 7 | Right Forearm |
| 8 | Right Hand |
| 9 | Left Upper Arm |
| 10 | Left Forearm |
| 11 | Left Hand |
| 12 | Right Thigh |
| 13 | Right Calf |
| 14 | Right Foot |
| 15 | Left Thigh |
| 16 | Left Calf |
| 17 | Left Foot |