**(Note: This file was generated with the in-game console command "dumpallclasses". Some of the listed commands may not work.)**

**Game Module Classes**

[**Actor**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Actor)**,** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate)**,** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity)**,** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave)**,** [**ScriptThread**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptThread)**,** [**Sentient**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Sentient)**,** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger)**,** [**World**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#World)

**Actor (*Actor*) ->** [**SimpleActor**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleActor) **->** [**Sentient**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Sentient) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**accuracy**

Set percent to hit

**accuracy**( *Float value* )

Set percent to hit

**accuracy**( *Float value* )

Set percent to hit

**ai\_off**

Turns the AI off for this actor.

**ai\_on**

Turns the AI on for this actor.

**aimat**( *String target* )

Specify the target to aim at.

**alarmnode**( *String value* )

Sets the name of the alarm node for the actor (must have type set to alarm for effect)

**alarmnode**( *String value* )

Sets the name of the alarm node for the actor (must have type set to alarm for effect)

**alarmnode**

Gets the name of the alarm node for the actor (must have type set to alarm for effect)

**alarmthread**( *String value* )

Sets the name of the alarm thread for the actor (must have type set to alarm for effect)

**alarmthread**( *String value* )

Sets the name of the alarm thread for the actor (must have type set to alarm for effect)

**alarmthread**

Gets the name of the alarm thread for the actor (must have type set to alarm for effect)

**ammo\_grenade**( *Integer grenade\_count* )

Gives the AI some grenades

**ammo\_grenade**

Returns how many grenades an AI has

**ammo\_grenade**( *Integer grenade\_count* )

Gives the AI some grenades

**anim**( *String name* )

Play animation.

**anim\_noclip**( *String name* )

Play noclip animation.

**anim\_scripted**( *String name* )

Play scripted animation.

**animfinal**

Whether the animation was succesfully finished

**animloop**( *String name* )

Loop animation.

**animname**

Sets the animname.

**animname**

Gets the animname.

**animscript**( *String name* )

Play the animation script

**animscript\_noclip**( *String name* )

Play the noclip animation script

**animscript\_scripted**( *String name* )

Play the scripted animation script

**attachgrenade**

Used only by grenade return animations to tell the code when to attach the grenade to the actor

**attackhandler**

Gets the current script that will handle attack events

**attackhandler**

Sets the current script that will handle attack events

**attackplayer**

Force Actor to attack the player

**avoidplayer**( *Integer allowavoid* )

set to 0 if this AI shouldn't automatically get out of the way, non-zero if he should.

**avoidplayer**

is 0 if this AI won't automatically get out of the way, non-zero if he will

**avoidplayer**( *Integer allowavoid* )

set to 0 if this AI shouldn't automatically get out of the way, non-zero if he should.

**balconyheight**

minimum height a balcony guy must fall to do special balcony death

**balconyheight**( *Float height* )

minimum height a balcony guy must fall to do special balcony death

**balconyheight**( *Float height* )

minimum height a balcony guy must fall to do special balcony death

**bedead**

Forces the actor to be instantly and totally dead; no death animation is played

**blendtime**

Set the crossblend time to something other than the default, in seconds

**blendtime**

Get the crossblend time

**breakspecial**

tell ai to break special attack

**calcgrenadetoss**( *Vector target\_position* )

Called to calculate a grenade toss. Must be called before a grenade throwing animation.  
Returns the name of the script to call with animscript if the toss can succeed, or if the toss won't work.  
Should be called infrequently, and never during the middle of a grenade toss.

**canmoveto**( *Vector position* )

returns a boolean if the AI can move to a point; for use in anim scripts

**cansee**( *Entity entity, [ Float fov ], [ Float vision\_distance ]* )

returns 1 if the entities can see eachother, 0 if not

**canshoot**( *Vector shootOrigin* )

Determines if it would be possible to shoot the sentient's enemy from the given position.

**canshootenemyfrom**( *Vector shootOrigin* )

Determines if it would be possible to shoot the sentient's enemy from the given position.

**crawlto**( *String dest* )

Specify the location to crawl to.

**crouchto**( *String dest* )

Specify the location to crouch to.

**damagepuff**( *Vector position, Vector direction* )

Spawns a puff of 'blood' smoke at the speficied location in the specified direction.

**deathembalm**

preps the dead actor for turning nonsolid gradually over time

**deathhandler**

Gets the current script that will handle death events

**deathhandler**

Sets the current script that will handle death events

**deathsinkstart**

Makes the entity sink into the ground and then get removed (this starts it).

**delete**

Removes this listener immediately.

**detachgrenade**

Used only by grenade return animations to tell the code when to throw the grenade

**disguise\_accept\_thread**( *String value* )

Sets the name of the thread for the actor to start when accepting papers

**disguise\_accept\_thread**

Gets the name of the thread for the actor to start when accepting papers

**disguise\_accept\_thread**( *String value* )

Sets the name of the thread for the actor to start when accepting papers

**disguise\_level**

Gets the disguise level of the actor. May be 1 or 2

**disguise\_level**( *Integer value* )

Sets the disguise level of the actor. May be 1 or 2

**disguise\_level**( *Integer value* )

Sets the disguise level of the actor. May be 1 or 2

**disguise\_period**( *Float period\_in\_seconds* )

Sets the time between the end of one disguise behavior and start of the next

**disguise\_period**( *Float period\_in\_seconds* )

Sets the time between the end of one disguise behavior and start of the next

**disguise\_period**

Gets the time between the end of one disguise behavior and start of the next

**disguise\_range**( *Float range\_in\_units* )

Sets the maximum distance for disguise behavior to get triggered

**disguise\_range**( *Float range\_in\_units* )

Sets the maximum distance for disguise behavior to get triggered

**disguise\_range**

Gets the maximum distance for disguise behavior to get triggered

**distancetoenemy**

Get the distance from the Actor to its enemy

**doActivate**( *Entity activatingEntity* )

General trigger event for all entities

**dumb**

Make Actor dumb.

**emotion**

The method of setting the facial expression of the Actor

**enableEnemy**

sets enableEnemy variable

**enablePain**

sets enablePain variable

**endactionanim**( )

End any aiming/action animation which is currently playing

**enemy**

Get the actor's current enemy

**enemy\_visible\_change\_time**

Get the last time whether or not the enemy is visible changed, in seconds

**enemysharerange**( *Float range* )

sets the range outside which the AI will not receive notification that a teammate has a new enemy

**enemysharerange**( *Float range* )

sets the range outside which the AI will not receive notification that a teammate has a new enemy

**enemysharerange**

gets the range outside which the AI will not receive notification that a teammate has a new enemy

**entitystart**

Initialize a Actor.

**eyeslookat**( *Entity entity* )

The actor will look at this entity.

**fallheight**( *Float height* )

Set the fallheight

**fallheight**

Set the fallheight

**favoriteenemy**

Gets this AI's favorite enemy

**favoriteenemy**( *Entity ai\_or\_player* )

Gets this AI's favorite enemy

**favoriteenemy**( *Entity ai\_or\_player* )

Gets this AI's favorite enemy

**fire\_grenade**

Used only by grenade throw animations to tell the code when to throw a grenade

**fixedleash**( *Float multiplier* )

if non-zero, the leash will never auto-reset; if zero, the leash may auto-reset

**fixedleash**( *Float multiplier* )

if non-zero, the leash will never auto-reset; if zero, the leash may auto-reset

**fixedleash**

if non-zero, the leash will never auto-reset; if zero, the leash may auto-reset

**fov**

The fov angle of the actor

**fov**( *Float angle* )

The fov angle of the actor

**fov**( *Float angle* )

The fov angle of the actor

**GetLocalYawFromVector**

Turn a worldspace vector into a local space yaw

**GetRunAnim**

Internal usage

**GetWalkAnim**

Internal usage

**gren\_awareness**

gets the awareness of grenades in 0-100 percent chance of responding to a grenadewhen the AI sees it (applied once every 0.4 seconds)

**gren\_awareness**( *Float awareness\_percent* )

sets the awareness of grenades in 0-100 percent chance of responding to a grenadewhen the AI sees it (applied once every 0.4 seconds)

**gren\_awareness**( *Float awareness\_percent* )

sets the awareness of grenades in 0-100 percent chance of responding to a grenadewhen the AI sees it (applied once every 0.4 seconds)

**gun**( *String s* )

specifies the gun to use

**gun**

gets the gun to being used

**gun**( *String s* )

specifies the gun to use

**hascompletelookahead**

returns true if there are no corners to turn on the rest of the AI's current path

**headmodel**( *String headmodel* )

sets the head model

**headmodel**

gets the head model

**headskin**

gets the head skin

**headskin**( *String headskin* )

sets the head skin

**hearing**( *Float radius* )

The hearing radius of the actor

**hearing**( *Float radius* )

The hearing radius of the actor

**hearing**( *Float radius* )

The hearing radius of the actor

**holster**

Holster weapon

**idlesay**( *String animation* )

The name of an idle dialog animation to play

**immediateremove**

Removes this listener immediately.

**inreload**

returns non-zero if the AI is in a reload

**inreload**( *Integer reloading* )

set to non-zero to indicate the AI is in a reload

**interrupt\_point**

hint from animation scripts to AI code that now is a good time to switch animations

**interval**( *Float distance* )

Sets the distance AI tries to keep between squadmates while moving.

**interval**( *Float distance* )

Sets the distance AI tries to keep between squadmates while moving.

**interval**

Gets the distance AI tries to keep between squadmates while moving.

**intervaldir**

the direction the AI would like to move to maintain its interval

**is\_enemy\_visible**

0 if the enemy is not currently visible, 1 if he is

**kickdir**

Gets the direction the AI wants to kick

**killed**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

event which is sent to an entity once it as been killed  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**last\_enemy\_visible\_time**

Get the last time the enemy was visible, in seconds

**leash**

Gets the maximum distance the AI will wander from its leash home

**leash**( *Float distance* )

Sets the maximum distance the AI will wander from its leash home

**leash**( *Float distance* )

Sets the maximum distance the AI will wander from its leash home

**lookaroundangle**( *Float angle* )

gets the angle in degrees left or right of center that the AI will look around while patrolling

**lookaroundangle**( *Float angle* )

gets the angle in degrees left or right of center that the AI will look around while patrolling

**lookaroundangle**

gets the angle in degrees left or right of center that the AI will look around while patrolling

**lookat**( *Entity entity* )

The actor will look at this entity.

**maxdist**

Gets the maximum distance the AI tries to keep between itself and the player

**maxdist**( *Float distance* )

Sets the maximum distance the AI tries to allow between itself and the player

**maxdist**( *Float distance* )

Sets the maximum distance the AI tries to allow between itself and the player

**mindist**( *Float distance* )

Sets the minimum distance the AI tries to keep between itself and the player

**mindist**

Gets the minimum distance the AI tries to keep between itself and the player

**mindist**( *Float distance* )

Sets the minimum distance the AI tries to keep between itself and the player

**mood**( *String new\_mood* )

sets the AI mood... must be 'bored', 'nervous', 'curious', or 'alert'.

**mood**

gets the AI mood: 'bored', 'nervous', 'curious', or 'alert'.

**movedir**

Returns a unit vector pointing in the current direction of motion, or zero if not moving.This still has meaning if velocity is zero but the AI is starting to move on a path.

**movedoneradius**( *Float radius* )

Set the waittill movedone radius, default 0 means don't use manual radius

**moveto**( *String anim, String dest* )

Specify the location to move to, with animation anim.

**mumble**( *Integer can\_mumble* )

Set to 1 if this guy is allowed to mumble, or 0 if he is not

**mumble**( *Integer can\_mumble* )

Set to 1 if this guy is allowed to mumble, or 0 if he is not

**mumble**

Returns 1 if this guy is allowed to mumble, or 0 if he is not

**no\_idle**

Gets if the actor will not go into idle after playing an animation

**no\_idle**

Specifies if the actor will not go into idle after playing an animation

**nolongpain**( *Integer allow* )

Set to 1 if long pain is not allowed, or 0 if long pain is allowed.

**nolongpain**

Returns 1 if long pain is not allowed, or 0 if long pain is allowed.

**nolongpain**( *Integer allow* )

Set to 1 if long pain is not allowed, or 0 if long pain is allowed.

**nosurprise**( *Integer nosurprise* )

set to 0 to allow this guy to play a surprised animation when first encountering an enemy.

**nosurprise**

gets whether or not this guy is allowed to play a surprised animation when first encountering an enemy.

**nosurprise**( *Integer nosurprise* )

set to 0 to allow this guy to play a surprised animation when first encountering an enemy.

**noticescale**( *Float multiplier* )

Set the max multiplier in time to notice an enemy (default 100, half as big notices twice as fast)

**noticescale**( *Float multiplier* )

Set the max multiplier in time to notice an enemy (default 100, half as big notices twice as fast)

**noticescale**

Get the max multiplier in time to notice an enemy (default 100, half as big notices twice as fast)

**pain**( *Entity attacker, Float damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

used to inflict pain to an entity

**painhandler**

Sets the current script that will handle pain events

**painhandler**

Gets the current script that will handle pain events

**pathdist**

returns total distance along current path to the path goal

**patrolpath**( *String value* )

Sets the name of the patrol path for the actor (must have type set to patrol for effect)

**patrolpath**

Gets the name of the patrol path for the actor (must have type set to patrol for effect)

**patrolpath**( *String value* )

Sets the name of the patrol path for the actor (must have type set to patrol for effect)

**physics\_off**

turn physics off.

**physics\_on**

turn physics on.

**playsound**( *String soundName, [ Integer channel ], [ Float volume ], [ Float min\_distance ], [ Float pitch ]* )

play a sound coming from this entity.  
default channel, CHAN\_BODY.

**pointat**( *Entity entity* )

The actor will point at this entity.

**position**

The Position the Actor wants to be and should animate towards

**position**

The Position the Actor wants to be and should animate towards

**ReadyToFire**

Returns if ready to fire

**reload\_mg42**

Reload the mg42 - only used by machinegunner

**remove**

Removes this listener the next time events are processed.

**resetleash**

resets the AI's leash to their current position

**runto**( *String dest* )

Specify the location to run to.

**say**( *String animation* )

The name of a dialog animation to play

**saydone**

Even used by sound-only special case of say to trigger waittill saydone

**setactionanim**( *String base\_anim, Float lower\_limit, Float upper\_limit* )

Set the base action animation and range that they cover

**setaimmotionanim**( *String anim\_crouch, String anim\_stand* )

Set aim motion animation (handler scripts only)

**setaimtarget**( *Entity entity* )

Sets the primary weapon's aim target.

**setanim**( *String anim, Integer slot, Float weight, String flagged* )

Set animation slot

**setanimlength**( *Float time* )

Set the maximum time an animation will play

**setmotionanim**( *String anim* )

Set motion animation (handler scripts only)

**setreloadcover**

do this command to let the ai know it needs to reload; used to reload while going to cover

**setsay**( *String animation* )

The name of a dialog animation to play - used by animation script only

**setupperanim**( *String anim* )

Set the upper body animation - used by animation script only

**share\_enemy**

internal code use only - shares an AI's enemy with his squad mates.

**share\_grenade**

internal code use only - shares an AI's grenade with his squad mates.

**sight**( *Float max\_sight\_range* )

Sets the vision distance of the actor.

**sight**

Gets the vision distance of the actor.

**sight**( *Float max\_sight\_range* )

Sets the vision distance of the actor.

**silent**( *Integer silent* )

set to 0 to prevent this guy from saying stuff besides pain and death sounds.

**silent**

gets whether or not this guy is allowed to say stuff besides pain and death sounds

**silent**( *Integer silent* )

set to 0 to prevent this guy from saying stuff besides pain and death sounds.

**sound\_awareness**

gets the awareness of sounds in 0-100 percent chance of hearing a sound withinhalf of the sound's radius' fades to zero outside sound's radius

**sound\_awareness**( *Float awareness\_percent* )

sets the awareness of sounds in 0-100 percent chance of hearing a sound withinhalf of the sound's radius' fades to zero outside sound's radius

**sound\_awareness**( *Float awareness\_percent* )

sets the awareness of sounds in 0-100 percent chance of hearing a sound withinhalf of the sound's radius' fades to zero outside sound's radius

**tether**( *Entity entity* )

the entity to which the AI's leash should be tethered

**thinkstate**

current ai think state; can be void, idle, pain, killed, attack, curious, disguise, or grenade.

**turndoneerror**

The error amount that turndone will occur for the turnto command.

**turndoneerror**( *Float error* )

The error amount that turndone will occur for the turnto command.

**turndoneerror**( *Float error* )

The error amount that turndone will occur for the turnto command.

**turnspeed**( *Float speed* )

The turn speed of the actor.

**turnspeed**

The turn speed of the actor.

**turnspeed**( *Float speed* )

The turn speed of the actor.

**turnto**( *Entity entity* )

The actor will turn to this entity.

**turret**

Gets the turret of the actor.

**turret**( *String turret* )

Sets the turret of the actor.

**turret**( *String turret* )

Sets the turret of the actor.

**type\_attack**

Gets the attack type of the actor.

**type\_attack**( *String value* )

Sets the attack type of the actor.

**type\_attack**( *String value* )

Sets the attack type of the actor.

**type\_disguise**

Gets the disguise type of the actor.

**type\_disguise**( *String value* )

Sets the disguise type of the actor.

**type\_disguise**( *String value* )

Sets the disguise type of the actor.

**type\_grenade**

Gets the grenade type of the actor.

**type\_grenade**( *String value* )

Sets the grenade type of the actor.

**type\_grenade**( *String value* )

Sets the grenade type of the actor.

**type\_idle**( *String value* )

Sets the idle type of the actor.

**type\_idle**( *String value* )

Sets the idle type of the actor.

**type\_idle**

Gets the idle type of the actor.

**unholster**

Unholster weapon

**upperanim**( *String anim* )

Set the upper body animation

**use**( *String name, [ Integer weapon\_hand ]* )

Use the specified weapon or item in the hand choosen (optional).

**voicetype**

Gets the voice type

**voicetype**

Set voicetype to magic letter postfix

**voicetype**

Set voicetype to magic letter postfix

**waittrigger**( *Boolean bool* )

If true, patrol guys and running men wait until triggered to move

**waittrigger**

If true, patrol guys and running men wait until triggered to move

**waittrigger**( *Boolean bool* )

If true, patrol guys and running men wait until triggered to move

**walkto**( *String dest* )

Specify the location to walk to.

**weapon**( *String weapon\_modelname* )

Gives the sentient the weapon specified.

**weapon**( *String weapon\_modelname* )

Sets the weapon.

**weapon**

Gets the weapon.

**weapon\_internal**( *String s* )

internal use

**weapongroup**

Specifies weapon animation set to use in anim scripts

**weapontype**

The Weapon Type of the Actor

**Ammo ->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**AmmoEntity ->** [**Item**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Item) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**ammoentity\_postspawn**

Ammo Entity Post Spawn

**Animate (*animate*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**anim**( *String animName* )

Exec anim commands on server or client.

**isloopinganim**( *String anim\_name* )

returns 1 if the anim is a looping anim, or 0 otherwise

**setcontrollerangles**( *Integer num, Vector angles* )

Sets the control angles for the specified bone.

**setsynctime**( *Float synctime* )

Set sync time for entity.

**AnimationEvent ->** [**Event**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Event) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**Archiver ->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**Armor ->** [**Item**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Item) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**BarrelObject (*func\_barrel*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**barreltype**( *String type* )

Sets the barrel's type

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**killed**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

event which is sent to an entity once it as been killed  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**Body ->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**Camera (*func\_camera*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**auto\_active**( *Boolean newActiveState* )

Whether or not the auto camera is active.

**auto\_maxfov**( *Float maxFOV* )

Sets the maximum FOV that should be used when automatically calculating FOV.

**auto\_radius**( *Float newRadius* )

Sets the radius of the automatic camera.

**auto\_starttime**( *Float newTime* )

Sets how long it takes for the camera to be switched to.

**auto\_state**( *String state1, [ String state2 ], [ String state3 ], [ String state4 ], [ String state5 ], [ String state6 ]* )

Sets the states the player needs to be in for this camera to activate.

**auto\_stoptime**( *Float newTime* )

Sets how long it takes for the camera switch back to the player.

**camera\_think**

Called each frame to allow the camera to adjust its position.

**continue**

Continue the camera movement.

**cut**

switch camera states immediately, do not transition

**doActivate**( *Entity activatingEntity* )

General trigger event for all entities

**fadetime**( *Float fadetime* )

Sets the fade time for camera transitioning.

**follow**( *Entity targetEnt, [ Entity targetWatchEnt ]* )

Makes the camera follow an entity and optionally watch an entity.

**follow\_distance**( *Float distance* )

Sets the camera follow distance.

**follow\_yaw**( *Float yaw* )

Sets the yaw offset of the camera following an entity.

**follow\_yaw\_absolute**

Makes the follow camera yaw absolute.

**follow\_yaw\_relative**

Makes the follow camera yaw relative (not absolute).

**fov**( *Float fov, [ Float fadeTime ]* )

Sets the camera's field of view (fov).  
if fadeTime is specified, camera will fade over that time  
if fov is less than 3, than an auto\_fov will be assumed  
the value of fov will be the ratio used for keeping a watch  
entity in the view at the right scale

**lookat**( *Entity ent* )

Makes the camera look at an entity.

**moveto**( *Entity ent* )

Move the camera's position to that of the specified entities.

**movetopos**( *Vector position* )

Move the camera's position to the specified position.

**nextcamera**( *String nextCamera* )

Sets the next camera to use.

**nowatch**( *[ Float fadeTime ]* )

Stop watching an entity or looking along a path.  
Camera is now static as far as orientation.  
if fadeTime is specified, camera will fade over that time

**orbit**( *Entity targetEnt, [ Entity targetWatchEnt ]* )

Makes the camera orbit around an entity and optionally watch an entity.

**orbit\_height**( *Float height* )

Sets the orbit camera's height.

**pause**

Pause the camera.

**speed**( *Float speed* )

Sets the camera speed.

**start**

Start camera moving.

**stop**

Stop the camera movement.

**turnto**( *Vector angle* )

Makes the camera look in the specified direction.

**watch**( *Entity watchEnt, [ Float fadeTime ]* )

Makes the camera watch an entity.  
if fadeTime is specified, camera will fade over that time

**watchnode**( *[ Float fadeTime ]* )

Makes the camera watch based on what is stored  
in the camera nodes.  
if fadeTime is specified, camera will fade over that time

**watchpath**( *[ Float fadeTime ]* )

Makes the camera look along the path of travel.  
if fadeTime is specified, camera will fade over that time

**CameraManager ->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**add**

Add a new point to the camera path where the player is standing.

**cancelFor**( *String name* )

Cancel for current path node event of type name

**delete**

Delete the current path node.

**hide**

Hides the paths.

**load**( *String filename* )

Loads a camera path.

**loop**( *[ Entity path ]* )

Loop the current path or the specified one.

**moveplayer**

Move the player to the current path node position.

**new**

Starts a new path.

**next**

Go to the next path node.

**nextpath**

Go to the next path.

**nowatch**

Set the current path node to watch nothing.

**play**( *[ Entity path ]* )

Play the current path or the specified one once.

**prev**

Go to the previous path node.

**prevpath**

Go to the previous path.

**renamepath**( *String newName* )

Rename the path to the new name.

**replace**

Replace the current path node position/angle with the player's.

**save**( *String filename* )

Saves the camera path.

**savemap**( *String filename* )

Saves the camera path to a map file.

**setfadetime**( *Float newFadeTime* )

Set the fadetime of the current path node.

**setfov**( *String newFOV* )

Set the fov at the current path node.

**setpath**( *Entity path* )

Sets the new path.

**setspeed**( *Float speed* )

Set the speed of the camera at the current path node.

**settarget**( *String target* )

Set the trigger target.

**settargetname**( *String targetname* )

Set the targetname.

**show**( *[ Entity path ]* )

Shows the specified path.

**stop**

Stop the camera playing path.

**updateinput**

Updates the current node with user interface values.

**waitTill**( *String name* )

Wait until current path node event of type name

**watch**( *String watch* )

Set the current path node to watch something.

**Class**

**ConsoleEvent ->** [**Event**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Event) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**CrateObject (*func\_crate*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**debristype**( *Integer type* )

Sets the debris type of the crate

**killed**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

event which is sent to an entity once it as been killed  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**Decal ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**DM\_Manager ->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**doroundtransition**

delayed function call to (possibly) determine round winner and restart next round

**finishroundtransition**

delayed function call to do the actual restart for the next round

**DM\_Team ->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**Door (*NormalDoor*) ->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**alwaysaway**

Makes the door always open away from the person opening it.

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**close**

Closes the door.

**dmg**( *Integer damage* )

Sets the amount of damage the door will do to entities that get stuck in it.

**doActivate**( *Entity activatingEntity* )

General trigger event for all entities

**doBlocked**( *Entity obstacle* )

sent to entity when blocked.

**door\_triggerfield**( *Entity other* )

Is called when a doors trigger field is touched.

**doorclosed**

Called when the door finishes closing.

**dooropened**

Called when the door finishes opening.

**doortype**( *String door\_type* )

Sets the defaults for this door  
Door Defaults: wood(default)  
metal

**doUse**( *Entity activatingEntity* )

sent to entity when it is used by another entity

**killed**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

event which is sent to an entity once it as been killed  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**linkdoor**

Link doors together.

**lock**

Lock the door.

**open**( *Entity other* )

Opens the door.

**sound\_close\_end**( *String sound\_close* )

Sets the sound to use when the door closes.

**sound\_close\_start**( *String sound\_close* )

Sets the sound to use when the door closes.

**sound\_locked**( *String sound\_locked* )

Sets the sound to use when the door is locked.

**sound\_message**( *String sound\_message* )

Sets the sound to use when the door displays a message.

**sound\_open\_end**( *String sound\_open\_end* )

Sets the sound to use when the door stops to opens.

**sound\_open\_start**( *String sound\_open\_start* )

Sets the sound to use when the door starts to opens.

**time**( *Float traveltime* )

Sets the time it takes for the door to open an close.

**toggledoor**( *Entity other* )

Toggles the state of the door (open/close).

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**trysolid**

Trys to make the door solid.

**tryToOpen**( *Entity other* )

Tries to open the door.

**unlock**

Unlock the door.

**wait**( *Float wait* )

Sets the amount of time to wait before automatically shutting.

**DrivableVehicle ->** [**Vehicle**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Vehicle) **->** [**VehicleBase**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#VehicleBase) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**doBlocked**( *Entity obstacle* )

sent to entity when blocked.

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**killed**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

event which is sent to an entity once it as been killed  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**DynItem ->** [**Item**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Item) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**kill**

console based command to kill yourself if stuck.

**EffectEntity (*effectentity*) ->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**Emitter (*func\_emitter*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**emitter**( *String name* )

Emitter to use

**idle**

Animates the puff daddy.

**Entity ->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**ai\_event**( *[ String type ], [ Float radius ]* )

Let the AI know that this entity made a sound,  
type is a string specifying what type of sound it is.  
radius determines how far the sound reaches.

**alpha**( *Float newAlpha* )

Set the alpha of the entity to alpha.

**alwaysdraw**

Sets this entity to always draw

**attach**( *Entity parent, String tagname, [ Integer use\_angles ]* )

attach this entity to the parent's legs tag called tagname

**attachedmodelanim**( *String tagname, String anim\_name, Float crossblend\_time, String model\_name* )

Tells models (or specified model) attached to specified tag to play   
specified animation. Crossbkend time doesn't work yet.

**attachmodel**( *String modelname, String tagname, [ Float scale ], [ String targetname ], [ Boolean detach\_at\_death ], [ Float removetime ], [ Float fadeintime ], [ Float fadeoutdelay ], [ Float fadetime ], [ Vector offset ]* )

attach a entity with modelname to this entity to tag called tagname.  
scale - scale of attached entities  
targetname - targetname for attached entities  
detach\_at\_death - when entity dies, should this model be detached.  
removetime - when the entity should be removed, if not specified, never.  
fadeintime - time to fade the model in over.  
fadeoutdelay - time to wait until we fade the attached model out  
fadeoutspeed - time the model fades out over  
offset - vector offset for the model from the specified tag

**avelocity**

gets the angular velocity for this entity.

**bind**( *Entity parent* )

bind this entity to the specified entity.

**brushmodel**

get the brush modelName.

**cansee**( *Entity entity, [ Float fov ], [ Float vision\_distance ]* )

returns 1 if the entities can see eachother, 0 if not

**censor**

used to ban certain contact when in parentmode

**classname**

The entity's classname

**classname**( *String nameOfClass* )

Determines what class to use for this entity,  
this is pre-processed from the BSP at the start  
of the level.

**connect\_paths**

Connects all navigation paths which intersect with the specified entity's volume

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**damage\_type**( *String meansofdeathstring* )

Set the type of damage that this entity can take

**deathsinkeachframe**

Makes the entity sink into the ground and then get removed (this gets called each frame).

**deathsinkstart**

Makes the entity sink into the ground and then get removed (this starts it).

**delete**

Removes this listener immediately.

**detach**

detach this entity from its parent.

**detachallchildren**

Detach all the children from the entity.

**disconnect\_paths**

Disconnects all navigation paths which intersect with the specified entity's volume

**droptofloor**( *[ Float maxRange ]* )

drops the entity to the ground, if maxRange is not specified 8192 is used.

**effects**( *[ String parameter1 ], [ String parameter2 ], [ String parameter3 ], [ String parameter4 ], [ String parameter5 ], [ String parameter6 ]* )

Change the current entity effects flags.  
Valid flags are as follows:  
+ sets a flag, - clears a flag  
antisbjuice - anti sucknblow juiceeveryframe - process commands every time entity is rendered

**entnum**

The entity's entity number

**explosionattack**( *String explosionModel, [ String tagName ]* )

Spawn an explosion optionally from a specific tag

**fade**( *[ Float fadetime<0.00...max\_float> ], [ Float target\_alpha<0.00...1.00> ]* )

Fade the entity's alpha, reducing it by 0.03  
every FRAMETIME, until it has faded out, does not remove the entity

**fadein**( *[ Float fadetime<0.00...max\_float> ], [ Float target\_alpha<0.00...1.00> ]* )

Fade the entity's alpha and scale in, increasing it by 0.03  
every FRAMETIME, until it has faded completely in to 1.0.  
If fadetime or target\_alpha are defined, they will override  
the default values.

**fadeout**( *[ Float fadetime<0.00...max\_float> ], [ Float alpha<0.00...1.00> ]* )

Fade the entity's alpha and scale out, reducing it by 0.03  
every FRAMETIME, until it has faded out. If fadetime or  
target\_alpha are defined, they will override the defaults.  
Once the entity has been completely faded, the entity is removed.

**flags**( *[ String parameter1 ], [ String parameter2 ], [ String parameter3 ], [ String parameter4 ], [ String parameter5 ], [ String parameter6 ]* )

Change the current entity flags.  
Valid flags are as follows:  
+ sets a flag, - clears a flag  
blood - should it bleed  
explode - should it explode when dead  
die\_gibs - should it spawn gibs when dead  
god - makes the entity invincible

**forceactivate**

Forces an entity to activate outside of the player's PVS

**getcontrollerangles**( *Integer num* )

Gets the control angles for the specified bone.

**gettagangles**( *String tag\_name* )

Gets the world angles of the tag

**gettagposition**( *String tag\_name* )

Gets the world position of the tag

**ghost**

make non-solid but still send to client regardless of hide status.

**glue**( *Entity parent, [ Integer glueAngles ]* )

glue this entity to the specified entity.

**gravity**( *Float gravityValue* )

Change the gravity on this entity

**heal**( *Float health* )

Adds health to an entity, 0-1 fraction of max\_health

**health**

entity's health

**health**( *Float newHealth* )

set the health (and max\_health) of the entity to newHealth

**health**( *Float newHealth* )

set the health (and max\_health) of the entity to newHealth

**healthonly**( *Float newHealth* )

set the health of the entity to newHealth without changing max\_health

**healthonly**( *Float newHealth* )

set the health of the entity to newHealth without changing max\_health

**hide**

hide the entity, opposite of show.

**hurt**( *Integer damage, [ String means\_of\_death ], [ Vector direction ]* )

Inflicts damage if the entity is damageable. If the number of damage  
points specified in the command argument is greater or equal than the  
entity's current health, it will be killed or destroyed.

**immediateremove**

Removes this listener immediately.

**immune**( *String immune\_string1, [ String immune\_string2 ], [ String immune\_string3 ], [ String immune\_string4 ], [ String immune\_string5 ], [ String immune\_string6 ]* )

Adds to the immunity list for this sentient.

**inpvs**( *Entity entity* )

returns 1 if the entities have connected pvs, 0 if not

**istouching**( *Entity entity* )

returns 1 if the entities are touching, 0 if not

**joinTeam**( *Entity teamMember* )

join a bind team.

**kill**

console based command to kill yourself if stuck.

**killattach**

kill all the attached entities.

**killed**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

event which is sent to an entity once it as been killed  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**killtarget**( *String targetName* )

when dying kill entities with this targetName.

**light**( *Float red, Float green, Float blue, Float radius* )

Create a dynmaic light on this entity.

**lightBlue**( *Float red* )

Set the red component of the dynmaic light on this entity.

**lightGreen**( *Float red* )

Set the red component of the dynmaic light on this entity.

**lightOff**

Turn the configured dynamic light on this entity off.

**lightOn**

Turn the configured dynmaic light on this entity on.

**lightRadius**( *Float red* )

Set the red component of the dynmaic light on this entity.

**lightRed**( *Float red* )

Set the red component of the dynmaic light on this entity.

**lightStyle**( *Integer lightStyleIndex* )

What light style to use for this dynamic light on this entity.

**loopsound**( *String soundName, [ Float volume ], [ String minimum\_distance ]* )

play a looped-sound with a certain volume and minimum\_distance  
which is attached to the current entity.

**mass**( *Float massAmount* )

set the mass of this entity.

**max\_health**

gets the entity's max health

**max\_health**( *Integer max\_health* )

sets max\_health without changing health

**max\_health**( *Integer max\_health* )

sets max\_health without changing health

**model**( *String modelName* )

set the model to modelName.

**model**( *String modelName* )

set the model to modelName.

**model**

get the modelName.

**movementstealth**( *Float scale* )

Sets the current movement stealth scalar for the sentient

**neverdraw**

Sets this entity to never draw

**nodamage**

entity does not take damage.

**normal\_damage**

The Normal Health of the Actor (0 - 100)

**normal\_health**

The Normal Health of the Actor

**normaldraw**

Sets this entity to normal draw

**notsolid**

make non-solid.

**playsound**( *String soundName, [ Integer channel ], [ Float volume ], [ Float min\_distance ], [ Float pitch ]* )

play a sound coming from this entity.  
default channel, CHAN\_BODY.

**pusher**( *Entity inflictor, Entity attacker, Vector direction, Float force* )

Push an entity in the specified direction with the specified force

**quitTeam**

quit the current bind team

**radnum**( *Float radnum* )

set the radnum

**radnum**

entity's radnum

**remove**

Removes this listener the next time events are processed.

**removeattachedmodel**( *String tagname, [ Float fadeRate ], [ String modelName ]* )

Removes the model attached to this entity at the specified tag.

**removeimmune**( *String immune\_string1, [ String immune\_string2 ], [ String immune\_string3 ], [ String immune\_string4 ], [ String immune\_string5 ], [ String immune\_string6 ]* )

Removes from the immunity list for this sentient.

**rendereffects**( *[ String parameter1 ], [ String parameter2 ], [ String parameter3 ], [ String parameter4 ], [ String parameter5 ], [ String parameter6 ]* )

Change the current render effects flags.  
Valid flags are as follows:  
+ sets a flag, - clears a flag  
dontdraw - send the entity to the client, but don't draw  
betterlighting - do sphere based vertex lighting on the entity  
lensflare - add a lens glow to the entity at its origin  
viewlensflare - add a view dependent lens glow to the entity at its origin  
lightoffset - use the dynamic color values as a light offset to the model  
skyorigin - this entity is the portal sky origin  
minlight - this entity always has some lighting on it  
fullbright - this entity is always fully lit  
additivedynamiclight - the dynamic light should have an additive effect  
lightstyledynamiclight - the dynamic light uses a light style, use the  
'lightstyle' command to set the index of the light style to be used

**rotatedbbox**( *Integer on\_off* )

Sets the entity's bbox to rotate with it.

**rotatedbbox**

Gets te entity's bbox to rotate with it.

**saydone**

Even used by sound-only special case of say to trigger waittill saydone

**scale**( *Float newScale* )

set the scale of the entity

**scale**( *Float newScale* )

set the scale of the entity

**scale**

get the scale of the entity

**scriptshader**( *String shaderCommand, Float argument1, [ Float argument2 ]* )

alias for shader command, change a specific shader parameter for the entity.  
Valid shader commands are:  
translation [trans\_x] [trans\_y] - change the texture translation  
offset [offset\_x] [offset\_y] - change the texture offset  
rotation [rot\_speed] - change the texture rotation speed  
frame [frame\_num] - change the animated texture frame  
wavebase [base] - change the base parameter of the wave function  
waveamp [amp] - change the amp parameter of the wave function  
wavebase [phase] - change the phase parameter of the wave function  
wavefreq [freq] - change the frequency parameter of the wave function

**setcontrollerangles**( *Integer num, Vector angles* )

Sets the control angles for the specified bone.

**setshaderdata**( *Float data0, Float data1* )

sets the shader controllers for this entity.

**setsize**( *Vector mins, Vector maxs* )

Set the bounding box of the entity to mins and maxs.

**shader**( *String shaderCommand, Float argument1, [ Float argument2 ]* )

change a specific shader parameter for the entity.  
Valid shader commands are:  
translation [trans\_x] [trans\_y] - change the texture translation  
offset [offset\_x] [offset\_y] - change the texture offset  
rotation [rot\_speed] - change the texture rotation speed  
wavebase [base] - change the base parameter of the wave function  
waveamp [amp] - change the amp parameter of the wave function  
wavebase [phase] - change the phase parameter of the wave function  
wavefreq [freq] - change the frequency parameter of the wave function

**show**

show the entity, opposite of hide.

**sighttrace**( *Vector start, Vector end, [ Integer pass\_entities ], [ Vector mins ], [ Vector maxs ]* )

Performs a trace line from the start to the end, returns 0 if something was hit and 1 otherwise

**solid**

make solid.

**spawnflags**( *Integer flags* )

spawnflags from the BSP, these are set inside the editor

**stationary**

entity does not move, causes no physics to be run on it.

**stoploopsound**

Stop the looped-sound on this entity.

**stopsound**( *[ Integer channel ]* )

stop the current sound on the specified channel.  
default channel, CHAN\_BODY.

**surface**( *String surfaceName, [ String parameter1 ], [ String parameter2 ], [ String parameter3 ], [ String parameter4 ], [ String parameter5 ], [ String parameter6 ]* )

change a legs surface parameter for the given surface.  
+ sets the flag, - clears the flag  
Valid surface commands are:  
skin1 - set the skin1 offset bit  
skin2 - set the skin2 offset bit  
nodraw - don't draw this surface

**svflags**( *[ String parameter1 ], [ String parameter2 ], [ String parameter3 ], [ String parameter4 ], [ String parameter5 ], [ String parameter6 ]* )

Change the current server flags.  
Valid flags are as follows:  
+ sets a flag, - clears a flag  
broadcast - always send this entity to the client

**takedamage**

makes entity take damage.

**team**( *String moveTeam* )

used to make multiple entities move together.

**touchtriggers**

this entity should touch triggers.

**trace**( *Vector start, Vector end, [ Integer pass\_entities ], [ Vector mins ], [ Vector maxs ]* )

Performs a Trace Line from the start to the end, returns the end or the position it hit at

**trigger**( *String name* )

Trigger the specified target or entity.

**unbind**

unbind this entity.

**unglue**

unglue this entity.

**usebbox**

do not perform perfect collision, use bounding box instead.

**velocity**

gets the velocity for this entity.

**velocity**( *Vector velocity* )

sets the velocity for this entity.

**volumedamage**( *Float damage* )

does damage to any entity within this's volume

**yaw**

entity's yaw

**Event ->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**ExplodeObject (*func\_explodeobject*) ->** [**MultiExploder**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#MultiExploder) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**amount**( *Integer amountOfDebris* )

How much debris to spawn each time.

**debrismodel**( *String debrisModel* )

What kind of debris to spawn when triggered.

**severity**( *Float newSeverity* )

How violently the debris should be ejected.

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**Exploder (*func\_exploder*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**dmg**( *Integer damage* )

Sets the damage the explosion does.

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**ExplodingWall (*func\_explodingwall*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**anglespeed**( *Float speed* )

Set the angle speed.

**base\_velocity**( *Vector velocity* )

Set the base velocity.

**checkonground**

Check if exploding wall is on ground.

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**dmg**( *Integer dmg* )

Set the damage from the exploding wall.

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**explosions**( *Integer explosions* )

Set the number of explosions.

**land\_angles**( *Vector angles* )

Set the land angles.

**land\_radius**( *Float radius* )

Set the land radius.

**random\_velocity**( *Vector velocity* )

Set the amount of random variation of the base velocity.

**stoprotating**

Stop rotating the wall.

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**Explosion ->** [**Projectile**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Projectile) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**constantdamage**

Makes the explosion do constant damage over the radius

**damageagain**

This event is generated each frame if explosion is set to damage each frame

**damageeveryframe**

Makes the explosion damage every frame

**explosioneffect**( *String explosionType* )

Make an explosionType explosion effect

**flash**( *Float time, Float r, Float g, Float b, Float radius* )

Flash player screens

**radius**( *Float projectileRadius* )

set the radius for the explosion

**radiusdamage**( *Float radiusDamage* )

set the radius damage an explosion does

**FallingRock (*func\_fallingrock*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**bounce**

sent to entity when touched.

**dmg**( *Integer dmg* )

Set the damage from the rock.

**doActivate**( *Entity activatingEntity* )

General trigger event for all entities

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**noise**( *String sound* )

Set the sound to play when the rock bounces

**rotate**

rotates the falling rock.

**speed**( *Float speed* )

Set the speed that the rock moves at.

**start**

Starts rock falling.

**wait**( *Float wait* )

How long to wait before rock starts falling.

**FileRead ->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**Flamethrower ->** [**Weapon**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Weapon) **->** [**Item**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Item) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**Fulcrum (*func\_fulcrum*) ->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**dampening**( *Float newDampening* )

dampening of fulcrum.

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**limit**( *Float newLimit* )

angular limit for the fulcrum.

**movesound**( *String newSinkSound* )

Sound played when fulcrum is moving.

**reset**

Reset the fulcrum right now.

**resetspeed**( *Float newResetspeed* )

Speed at which fulcrum resets itself, defaults to 0.002 \* speed.

**speed**( *Float speed* )

Speed at which fulcrum operates itself.

**FuncBeam (*func\_beam*) ->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**activate**

Activate the beam

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**angles**( *Vector newAngles<0.00...360.00><0.00...360.00><0.00...360.00>* )

set the angles of the entity to newAngles.

**color**( *Vector beam\_color<0.00...1.00><0.00...1.00><0.00...1.00>* )

Set the color of the beam

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**deactivate**

Deactivate the beam

**delay**( *Float delay* )

Set the amount of delay on the beam updater

**doActivate**( *Entity activatingEntity* )

General trigger event for all entities

**endalpha**( *Float alpha* )

Set the endpoint alpha value of the beam

**endpoint**( *Vector beam\_end\_point* )

Set the end point of the beam. The beam will be draw from the origin to  
the end point.

**findendpoint**

Find the endpoint of a beam

**life**( *Float beam\_life* )

Set the amount of time the beam stays on when activated

**maxoffset**( *Float max\_offset* )

Set the maximum offset the beam can travel above, below, forward or back of it's endpoints

**minoffset**( *Float min\_offset* )

Set the minimun offset the beam can travel above, below, forward or back of it's endpoints

**model**( *String modelName* )

set the model to modelName.

**numsegments**( *Integer numsegments* )

Set the number of segments for the beam

**numspherebeams**( *Integer num* )

Set the number of beams that will be shot out in a sphere like formation

**overlap**( *Float beam\_overlap* )

Set the amount of overlap the beams have when they are being strung together

**persist**( *Boolean bool* )

Set the persist property of the beam

**radius**( *Float radius* )

Set the starting radius of the beams if this is a beamsphere

**shader**( *String beam\_shader* )

Set the shader that the beam will use

**shoot**

Make the beam cause damage to entities that get in the way

**shootradius**( *Float radius* )

Set the radius of the damage area between beam endpoints

**target**( *String beam\_target* )

Set the target of the beam. The beam will be drawn from the origin  
to the origin of the target entity

**tileshader**( *String beam\_shader* )

Set the shader that the beam will use. This shader will be tiled.

**toggledelay**( *[ String [random] ], [ Float time ]* )

Causes a beam toggling effect. Sets the time between toggling. If random is specified, The time will be between 0 and time

**updateendpoint**

Update the endpoint of a beam

**updateorigin**

Update the origin of a beam

**FuncLadder (*func\_ladder*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**FuncRemove (*func\_remove*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**Game ->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**detail**

game.detail

**skill**

game.skill

**Gib (*gib*) ->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**stopped**

sent when entity has stopped bouncing for MOVETYPE\_TOSS.

**throwgib**( *Entity ent, Integer velocity, Float scale* )

Throw a gib.

**GrenadeHint (*info\_grenadehint*) ->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**Health (*health\_020*) ->** [**Item**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Item) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**health\_postspawn**

Health Post Spawn

**item\_pickup**( *Entity item* )

Pickup the specified item.

**HelmetObject (*helmetobject*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**HorizontalPipe (*func\_horizontalpipe*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**InfoNotNull (*info\_notnull*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**InfoNull (*info\_null*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**InteractObject (*interactobject*) ->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**hiteffect**

Sets the tiki it will spawn when it's hit

**killed**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

event which is sent to an entity once it as been killed  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**killedeffect**

Sets the tiki it will spawn when it's destroyed

**InventoryItem ->** [**Weapon**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Weapon) **->** [**Item**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Item) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**activateitem**( *[ String mode ]* )

Activates the item

**activatepapers**

The activation of the papers item

**Item ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**amount**( *Integer amount* )

Sets the amount of the item.

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**dialogneeded**( *String dialog\_needed* )

Sets the dialog needed string.

**dmamount**( *Integer amount* )

Sets the amount of the item for DM.

**dmmaxamount**( *Integer max\_amount* )

Sets the max amount of the item for DM.

**doUse**( *Entity activatingEntity* )

sent to entity when it is used by another entity

**item\_droptofloor**

Drops the item to the ground.

**item\_pickup**( *Entity item* )

Pickup the specified item.

**maxamount**( *Integer max\_amount* )

Sets the max amount of the item.

**name**( *String item\_name* )

Sets the item name.

**no\_remove**

Makes it so the item is not removed from the world when it is picked up.

**pickup\_done**

Called when the item pickup is done.

**pickupsound**( *String name* )

sets the item's pickup sound alias

**respawn**

Respawns the item.

**respawn\_done**

Called when the item respawn is done.

**respawnsound**

Turns on the respawn sound for this item.

**set\_respawn**( *Integer respawn* )

Turns respawn on or off.

**set\_respawn\_time**( *Float respawn\_time* )

Sets the respawn time.

**stopped**

sent when entity has stopped bouncing for MOVETYPE\_TOSS.

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**Level ->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**alarm**( *Integer alarm\_status* )

zero = global level alarm off, non-zero = alarm on

**alarm**

zero = global level alarm off, non-zero = alarm on

**bombs\_planted**

Gets the number of bombs that are set

**bombs\_planted**( *Integer num* )

Sets the number of bombs that are set

**bombs\_planted**( *Integer num* )

the number of bombs that are set

**clockside**( *String axis\_allies\_draw\_kills* )

Sets which side the clock is on... 'axis' or 'allies' win when time is up, 'kills' gives the win to the team with more live members, 'draw' no one wins

**clockside**

Gets which side the clock is on... 'axis' or 'allies' win when time is up

**clockside**( *String axis\_or\_allies* )

Sets which side the clock is on... 'axis' or 'allies' win when time is up

**dmrespawning**( *Integer allow\_respawn* )

set to 1 to turn on wave-based DM, to 0 to disable respawns within a round

**dmrespawning**( *Integer allow\_respawn* )

set to 1 to turn on wave-based DM, to 0 to disable respawns within a round

**dmrespawning**

returns 1 if wave-based DM, 0 if respawns are disabled within a round

**dmroundlimit**( *Integer roundlimit* )

sets the default roundlimit, in minutes; can be overridden by 'roundlimit' cvar

**dmroundlimit**( *Integer roundlimit* )

sets the default roundlimit, in minutes; can be overridden by 'roundlimit' cvar

**dmroundlimit**

gets the actual roundlimit, in minutes; may be 'roundlimit' cvar or the default round limit

**found\_secrets**

count of found secrets

**loop\_protection**

says if infinite loop protection is enabled

**loop\_protection**( *Integer loop\_protection* )

specify if infinite loop protection is enabled

**nodrophealth**( *Integer alarm\_status* )

zero = automatically drop health according to cvars, non-zero = don't autodrop health (like hard mode)

**papers**

the level of papers the player currently has

**papers**

the level of papers the player currently has

**planting\_team**

Gets which is planting the bomb, 'axis' or 'allies'

**planting\_team**( *String axis\_or\_allies* )

Sets which is planting the bomb, 'axis' or 'allies'

**planting\_team**( *String axis\_allies\_draw\_kills* )

which is planting the bomb, 'axis' or 'allies'

**prespawnsentient**

internal usage.

**rain\_density**

Sets the rain density

**rain\_density**

Sets the rain density

**rain\_length**

Sets the rain length

**rain\_length**

Sets the rain length

**rain\_min\_dist**

Sets the rain min\_dist

**rain\_min\_dist**

Sets the rain min\_dist

**rain\_numshaders**

Sets the rain numshaders

**rain\_numshaders**

Sets the rain numshaders

**rain\_shader**

Sets the rain shader

**rain\_shader**

Sets the rain shader

**rain\_slant**

Sets the rain slant

**rain\_slant**

Sets the rain slant

**rain\_speed**

Sets the rain speed

**rain\_speed**

Sets the rain speed

**rain\_speed\_vary**

Sets the rain speed variance

**rain\_speed\_vary**

Sets the rain speed variance

**rain\_width**

Sets the rain width

**rain\_width**

Sets the rain width

**roundbased**

Gets wether or not the game is currently round based or not

**targets\_destroyed**( *Integer num* )

the number of bomb targets that have been destroyed

**targets\_destroyed**( *Integer num* )

Sets the number of bomb targets that have been destroyed

**targets\_destroyed**

Gets the number of bomb targets that have been destroyed

**targets\_to\_destroy**( *Integer num* )

the number of bomb targets that must be destroyed

**targets\_to\_destroy**( *Integer num* )

Sets the number of bomb targets that must be destroyed

**targets\_to\_destroy**

Gets the number of bomb targets that must be destroyed

**time**

current level time

**total\_secrets**

count of total secrets

**Light (*light*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**LightStyleClass ->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**Listener ->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**cancelFor**( *String name* )

Cancel for event of type name

**classname**

classname variable

**commanddelay**( *Float delay, String command, [ String [arg1] ], [ String [arg2] ], [ String [arg3] ], [ String [arg4] ], [ String [arg5] ], [ String [arg6] ]* )

executes a command after the given delay

**delaythrow**( *String label* )

Internal usage.

**delete**

Removes this listener immediately.

**exec**( *String script* )

Executes the specified script.

**exec**( *String script* )

Executes the specified script.

**immediateremove**

Removes this listener immediately.

**owner**

Returns the owner.

**remove**

Removes this listener the next time events are processed.

**thread**( *String label* )

Creates a thread starting at label.

**thread**( *String label* )

Creates a thread starting at label.

**throw**( *String label* )

Throws to the specified label.

**unregister**( *String label* )

Unregisters the label from the event of the same name.

**waitexec**( *String script* )

Executes the specified script and waits until the called thread group is finished.

**waitexec**( *String script* )

Executes the specified script and waits until the called thread group is finished.

**waitthread**( *String label* )

Creates a thread starting at label and waits until the called thread is finished.

**waitthread**( *String label* )

Creates a thread starting at label and waits until the called thread is finished.

**waitTill**( *String name* )

Wait until event of type name

**LODMaster (*lodmaster*) ->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**lod\_angles**( *Float pitch<0.00...360.00>, Float yaw<0.00...360.00>, Float roll<0.00...360.00>* )

Set the angles of the model

**lod\_discard**

Discard LOD changes

**lod\_roll**( *Float roll* )

Set the roll of the model

**lod\_spawn**( *String model* )

Edit LOD on the specified model

**lod\_yaw**( *Float yaw* )

Set the yaw of the model

**LODSlave (*lodslave*) ->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**lod\_discard**

Discard LOD changes

**lod\_model**

Set the model of the current viewthing

**lod\_think**

Called every frame to process the view thing.

**MonkeyBars (*func\_monkeybars*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**Mover (*mover*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**movedone**( *Entity finishedEntity* )

Sent to commanding thread when done with move .

**MultiExploder (*func\_multi\_exploder*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**dmg**( *Integer damage* )

Sets the damage the explosion does.

**duration**( *Float duration* )

Sets the duration of the explosion.

**random**( *Float randomness* )

Sets the randomness value of the explosion.

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**wait**( *Float explodewait* )

Sets the wait time of the explosion.

**Object (*object*) ->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**handlespawn**

Internal usage

**model**( *String modelName* )

set the model to modelName.

**model**( *String modelName* )

set the model to modelName.

**Parm ->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**motionfail**

motionfail

**movedone**

movedone

**movefail**

movefail

**other**

other

**owner**

owner

**previousthread**

Returns the previous thread.

**sayfail**

sayfail

**upperfail**

upperfail

**PathNode (*info\_pathnode*) ->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**delete**

Removes this listener immediately.

**immediateremove**

Removes this listener immediately.

**istouching**( *Entity entity* )

returns 1 if the entities are touching, 0 if not

**spawnflags**( *Integer node\_flags* )

Sets the path nodes flags.

**PathSearch ->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**Player (*player*) ->** [**Sentient**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Sentient) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**actionanimend**

Called when the action animation has finished.

**actor**( *String modelname, [ String keyname1 ], [ String value1 ], [ String keyname2 ], [ String value2 ], [ String keyname3 ], [ String value3 ], [ String keyname4 ], [ String value4 ]* )

Spawns an actor.

**actorinfo**( *Integer actor\_number* )

Prints info on the specified actor.

**animloop\_legs**

Called when the legs animation has finished.

**animloop\_torso**

Called when the torso animation has finished.

**attachtoladder**

Attaches the sentient to a ladder

**auto\_join\_team**

Join the team with fewer players

**callvote**( *String arg1, [ String arg2 ]* )

Player calls a vote

**coord**

Prints out current location and angles

**correctweaponattachments**

makes sure the weapons is properly attached when interupting a reload

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**damage\_multiplier**( *Float damage\_multiplier* )

Sets the current damage multiplier

**dead**

Called when the player is dead.

**deadbody**

Spawn a dead body

**dmdeathdrop**

Drops the player inventory in DM after's he's been killed

**dmmessage**( *Integer mode, String stuffstrings* )

sends a DM message to the appropriate players

**dmteam**

returns 'allies', 'axis', 'spectator', or 'freeforall'

**dog**( *[ Integer god\_mode ]* )

Sets the god mode cheat or toggles it.

**endlevel**

Called when the player gets to the end of the level.

**ensureforwardoffladder**

Ensures that the player went forward off the ladder.

**ensureoverladder**

Ensures that the player is at the proper height when getting off the top of a ladder

**enter**( *Entity vehicle, [ String driver\_anim ]* )

Called when someone gets into a vehicle.

**exit**( *Entity vehicle* )

Called when driver gets out of the vehicle.

**face**( *Vector angles* )

Force angles to specified vector

**finishuseanim**

Fires off all targets associated with a particular useanim.

**finishuseobject**

Fires off all targets associated with a particular useobject.

**fireheld**

returns 1 if this player is holding fire, or 0 if he is not

**forcelegsstate**( *String legsstate* )

Force the player's legs to a certain state

**forcetorsostate**( *String torsostate* )

Force the player's torso to a certain state

**fov**( *[ Float fov ]* )

Sets the player's fov.

**fullheal**

Heals player.

**gameversion**

Prints the game version.

**getmovement**

returns the player current movement

**getposition**

returns the player current position

**give**( *String name, [ Integer amount ]* )

Gives the player the specified thing (weapon, ammo, item, etc.) and optionally the amount.

**giveweapon**( *String weapon\_name* )

Gives the player the specified weapon.

**gotkill**( *Entity victim, Integer damage, Entity inflictor, Integer meansofdeath, Boolean gib* )

event sent to attacker when an entity dies

**has\_disguise**( *Integer is\_disguised* )

zero = does not have a disguise, non-zero = has a disguise

**has\_disguise**

zero = does not have a disguise, non-zero = has a disguise

**holster**

Holsters all wielded weapons, or unholsters previously put away weapons

**invnext**

Cycle to player's next item.

**invprev**

Cycle to player's previous item.

**iprint**( *String string, [ Integer bold ]* )

prints a string to the player, optionally in bold

**is\_disguised**

zero = not disguised, non-zero = disguised

**join\_team**( *String team* )

Join the specified team (allies or axis)

**jump**( *Float height* )

Makes the player jump.

**jumpxy**( *Float forwardmove, Float sidemove, Float speed* )

Makes the sentient jump.

**kill**

console based command to kill yourself if stuck.

**killclass**( *String classname, [ Integer except\_entity\_number ]* )

Kills all of the entities in the specified class.

**killed**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

event which is sent to an entity once it as been killed  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**killent**( *Integer entity\_number* )

Kills the specified entity.

**leave\_team**

Leave the current team

**listinventory**

List of the player's inventory.

**logstats**( *Boolean state* )

Turn on/off the debugging playlog

**modheight**( *String height* )

change the maximum height of the player  
can specify 'stand', 'duck', 'duckrun', 'prone', or a specific height

**moveposflags**( *String position, [ String movement ]* )

used by the state files to tell the game dll what the player is doing

**nextpaintime**( *Float seconds* )

Set the next time the player experiences pain (Current time + seconds specified).

**noclip**

Toggles the noclip cheat.

**notarget**

Toggles the notarget cheat.

**notready**

makes this player not ready for the round to start

**objectivecount**( *Integer num\_completed, Integer out\_of* )

Sets the number of objectives completed and the total number of objectives

**pain**( *Entity attacker, Float damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

used to inflict pain to an entity

**perferredweapon**( *String weapon\_name* )

Overrides your perferred weapon that is displayed in the stats screen.

**physics\_off**

turn player physics off.

**physics\_on**

turn player physics on.

**poweruptimer**( *Integer poweruptimer, Integer poweruptype* )

Sets the powerup timer and powerup type.

**primarydmweapon**( *String weaptype* )

Sets the player's primary DM weapon

**ready**

makes this player ready for the round to start

**ready**

returns 1 if this player is ready, 0 otherwise

**reload**

Reloads the player's weapon

**removeclass**( *String classname, [ Integer except\_entity\_number ]* )

Removes all of the entities in the specified class.

**removeent**( *Integer entity\_number* )

Removes the specified entity.

**resethaveitem**( *String weapon\_name* )

Resets the game var that keeps track that we have gotten this weapon

**resetstate**

Reset the player's state table.

**respawn**

Respawns the player.

**safeholster**( *Boolean putaway* )

Holsters all wielded weapons, or unholsters previously put away weapons  
preserves state, so it will not holster or unholster unless necessary

**safezoom**( *Boolean zoomin* )

0 turns off zoom, and 1 returns zoom to previous setting

**score**

Show the score for the current deathmatch game

**show**

show the entity, opposite of hide.

**skipcinematic**

Skip the current cinematic

**spawn**( *String entityname, [ String keyname1 ], [ String value1 ], [ String keyname2 ], [ String value2 ], [ String keyname3 ], [ String value3 ], [ String keyname4 ], [ String value4 ]* )

Spawns an entity.

**spectator**

Become a spectator

**startuseobject**

starts up the useobject's animations.

**state**

Dumps the player's state to the console.

**stats**

Display the MissionLog.

**stopwatch**( *Integer duration* )

Starts a stopwatch for a given duration... use 0 to clear the stopwatch

**stopwatchingactor**( *Entity actor\_to\_stop\_watching* )

Makes the player's camera stop watching the specified actor.

**stufftext**( *String stuffstrings* )

Stuffs text to the player's console

**takepain**( *Boolean bool* )

Set whether or not to take pain

**tele**( *Vector location* )

Teleport to location

**testthread**( *String scriptfile, [ String label ]* )

Starts the named thread at label if provided.

**turn**( *Float yawangle* )

Causes player to turn the specified amount.

**turnlegs**( *Float yawangle* )

Turns the players legs instantly by the specified amount.

**turnupdate**( *Float yaw, Float timeleft* )

Causes player to turn the specified amount.

**turretenter**( *Entity turret, [ String driver\_anim ]* )

Called when someone gets into a turret.

**turretexit**( *Entity turret* )

Called when driver gets out of the turret.

**tweakladderpos**

Tweaks the player's position on a ladder to be proper

**unattachfromladder**

Unattaches the sentient from a ladder

**updatepoweruptime**

Called once a second to decrement powerup time.

**useheld**

returns 1 if this player is holding use, or 0 if he is not

**usestuff**

Makes the player try to use whatever is in front of her.

**viewangles**

get the angles of the entity.

**viewangles**( *Vector newAngles* )

set the view angles of the entity to newAngles.

**viewmodelanim**( *String anim, [ Integer force\_restart ]* )

Sets the player's view model animation

**voicetype**( *String voice\_name* )

Sets the voice type to use the player.

**vote**( *String arg1* )

Player votes either yes or no

**waitForState**( *String stateToWaitFor* )

When set, the player will clear waitforplayer when this state is hit  
in the legs or torso.

**watchactor**( *Entity actor\_to\_watch* )

Makes the player's camera watch the specified actor.

**weapdrop**

Drops the player's current weapon.

**weapnext**

Cycle to player's next weapon.

**weapprev**

Cycle to player's previous weapon.

**whatis**( *Integer entity\_number* )

Prints info on the specified entity.

**wuss**

Gives player all weapons.

**zoomoff**

makes sure that zoom is off

**PlayerAlliedDeathmatchStart (*info\_player\_allied*) ->** [**PlayerDeathmatchStart**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#PlayerDeathmatchStart) **->** [**PlayerStart**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#PlayerStart) **->** [**SimpleArchivedEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleArchivedEntity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**PlayerAxisDeathmatchStart (*info\_player\_axis*) ->** [**PlayerDeathmatchStart**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#PlayerDeathmatchStart) **->** [**PlayerStart**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#PlayerStart) **->** [**SimpleArchivedEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleArchivedEntity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**PlayerDeathmatchStart (*info\_player\_deathmatch*) ->** [**PlayerStart**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#PlayerStart) **->** [**SimpleArchivedEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleArchivedEntity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**arena**( *Integer arena\_number* )

set the arena number for this starting position

**PlayerIntermission (*info\_player\_intermission*) ->** [**Camera**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Camera) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**PlayerStart (*info\_player\_start*) ->** [**SimpleArchivedEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleArchivedEntity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**disablespawn**

forbids spawning from this spawnpoint

**enablespawn**

allows spawning from this spawnpoint

**PortalCamera (*portal\_camera*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**roll**( *Float roll* )

Sets the portal camera's roll.

**PortalSurface (*portal\_surface*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**locatecamera**

Locates the camera position.

**Projectile (*projectile*) ->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**addownervelocity**( *Boolean bool* )

Set whether or not the owner's velocity is added to the projectile's velocity

**addvelocity**( *Float velocity\_x, Float velocity\_y, Float velocity\_z* )

Set a velocity to be added to the projectile when it is created

**avelocity**( *[ String [random|crandom] ], [ Float yaw ], [ String [random|crandom] ], [ Float pitch ], [ String [random|crandom] ], [ Float roll ]* )

set the angular velocity of the projectile

**beam**( *String command, [ String arg1 ], [ String arg2 ], [ String arg3 ], [ String arg4 ], [ String arg5 ], [ String arg6 ]* )

send a command to the beam of this projectile

**bouncesound**

Set the name of the sound that is played when the projectile bounces

**bouncesound\_hard**

Set the name of the sound that is played when the projectile bounces off hard surfaces

**bouncesound\_metal**

Set the name of the sound that is played when the projectile bounces off metal

**bouncesound\_water**

Set the name of the sound that is played when the projectile bounces in water

**bouncetouch**

Make the projectile bounce when it hits a non-damageable solid

**canhitowner**

Make the projectile be able to hit its owner

**chargelife**

set the projectile's life to be determined by the charge time

**chargespeed**

set the projectile's speed to be determined by the charge time

**clearowner**

Make the projectile be able to hit its owner now

**dlight**( *Float red, Float green, Float blue, Float intensity* )

set the color and intensity of the dynamic light on the projectile

**dmlife**( *Float projectileLife* )

set the life of the projectile in DM

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**drunk**( *Float amount, Float rate* )

Make the projectile drunk

**explode**

Make the projectile explode

**explodeontouch**

Make the projectile explode when it touches something damagable

**explosionmodel**( *String modelname* )

Set the modelname of the explosion to be spawned

**heatseek**

Make the projectile heat seek

**hitdamage**( *Float projectileHitDamage* )

set the damage a projectile does when it hits something

**impactmarkorientation**( *Float degrees* )

Set the orientation of the impact mark

**impactmarkradius**( *Float radius* )

Set the radius of the impact mark

**impactmarkshader**( *String shader* )

Set the impact mark of the shader

**killed**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

event which is sent to an entity once it as been killed  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**knockback**( *Float projectileKnockback* )

set the knockback of the projectile when it hits something

**life**( *Float projectileLife* )

set the life of the projectile

**meansofdeath**( *String meansOfDeath* )

set the meansOfDeath of the projectile

**minlife**( *Float minProjectileLife* )

set the minimum life of the projectile (this is for charge up weapons)

**minspeed**( *Float minspeed* )

set the minimum speed of the projectile (this is for charge up weapons)

**notouchdamage**

Makes the projectile not blow up or deal damage when it touches a damagable object

**prethink**

Make the projectile think to update it's velocity

**removewhenstopped**

Make the projectile get removed when it stops

**smashthroughglass**( *Integer speed* )

Makes the projectile smash through windows & other damageble glass objects when going above a set speed

**speed**( *Float projectileSpeed* )

set the speed of the projectile

**stopped**

sent when entity has stopped bouncing for MOVETYPE\_TOSS.

**updatebeam**

Update the attached beam

**PuffDaddy (*plant\_puffdaddy*) ->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**idle**

Animates the puff daddy.

**PushObject (*func\_pushobject*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**dmg**( *Integer damage* )

Set the damage.

**doBlocked**( *Entity obstacle* )

sent to entity when blocked.

**pushsound**( *String sound* )

Set the pushing sound

**start**

Sets up the pushobject.

**Rain (*func\_rain*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**RandomSpawn (*func\_randomspawn*) ->** [**Spawn**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Spawn) **->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**doActivate**( *Entity activatingEntity* )

General trigger event for all entities

**max\_time**( *Float maxTime* )

Maximum time between random spawns.

**min\_time**( *Float minTime* )

Minimum time between random spawns.

**RandomSpeaker (*sound\_randomspeaker*) ->** [**TriggerSpeaker**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#TriggerSpeaker) **->** [**TriggerPlaySound**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#TriggerPlaySound) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**chance**( *Float newChance<0.00...1.00>* )

Sets the chance that the sound will play when triggered.

**maxdelay**( *Float max\_delay* )

Sets the maximum time between playings.

**mindelay**( *Float min\_delay* )

Sets the minimum time between playings.

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**ReSpawn (*func\_respawn*) ->** [**Spawn**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Spawn) **->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**RotatingDoor (*func\_rotatingdoor*) ->** [**Door**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Door) **->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**doclose**

Closes the door (special doors).

**doopen**( *Entity other* )

Opens the door (special doors).

**openangle**( *Float open\_angle* )

Sets the open angle of the door.

**RunThrough (*func\_runthrough*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**chance**( *Float chance* )

chance that trigger will spawn something.

**delay**( *Float delay* )

time between RunThrough being activated.

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**lip**( *Float lip* )

distance below trigger we should spawn things.

**offset**( *Vector spawn\_offset* )

When triggered, what to offset the spawned object by.

**spawnmodel**( *String model\_to\_spawn* )

When triggered, what to spawn.

**speed**( *Float speed* )

threshold speed at which RunThrough is activated.

**Script ->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**ScriptClass ->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**ScriptDoor (*script\_door*) ->** [**Door**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Door) **->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**doclose**

Closes the door (special doors).

**doinit**

Sets up the script door.

**doopen**( *Entity other* )

Opens the door (special doors).

**movedir**

door's movedir

**opendot**

door's open dot product

**startangles**

door's size

**startorigin**

door's startorigin

**ScriptMaster ->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**alias**( *String alias, String realPath, [ String arg1 ], [ String arg2 ], [ String arg3 ], [ String arg4 ], [ String arg5 ], [ String arg6 ]* )

Create an alias to the specified path

**aliascache**( *String alias, String realPath, [ String arg1 ], [ String arg2 ], [ String arg3 ], [ String arg4 ], [ String arg5 ], [ String arg6 ]* )

Create an alias to the specified path and cache the resource

**cache**( *String resourceName* )

pre-cache the given resource.

**ScriptModel (*script\_model*) ->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**anim**( *String anim\_name* )

Sets the script model's animation

**model**( *String modelName* )

set the model to modelName.

**ScriptOrigin (*script\_origin*) ->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**model**( *String modelName* )

set the model to modelName.

**ScriptSkyOrigin (*script\_skyorigin*) ->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**ScriptSlave (*script\_object*) ->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**angles**( *Vector angles* )

Sets the angles.

**bind**( *Entity parent* )

bind this entity to the specified entity.

**closeportal**

Close the area portal enclosed in this object

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**dmg**( *Integer damage* )

Set the damage.

**doActivate**( *Entity activatingEntity* )

General trigger event for all entities

**doBlocked**( *Entity obstacle* )

sent to entity when blocked.

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**doUse**( *Entity activatingEntity* )

sent to entity when it is used by another entity

**endpath**

Stop following the path

**explode**( *Float damage* )

Creates an explosion at the script slave's position

**flypath**( *Entity array, Float speed, Float acceleration, Float look\_ahead* )

Makes the script slave fly the specified path with speed and acceleration until reached\_distance close to position

**followpath**( *Entity path, [ String arg1 ], [ String arg2 ], [ String arg3 ], [ String arg4 ], [ String arg5 ], [ String arg6 ]* )

Makes the script slave follow the specified path. The allowable arguments are ignoreangles,  
ignorevelocity, normalangles, loop, and a number specifying the start time.

**jumpto**( *String vector\_or\_entity* )

Jump to specified vector or entity.

**model**( *String modelName* )

set the model to modelName.

**modifyflypath**( *Float speed, Float acceleration, Float look\_ahead* )

Makes the script slave fly the specified path with speed and acceleration until reached\_distance close to position

**move**

Move the script slave.

**moveBackward**( *Float dist* )

Move the position backward.

**moveDown**( *Float dist* )

Move the position down.

**moveEast**( *Float dist* )

Move the position east.

**moveForward**( *Float dist* )

Move the position forward.

**moveLeft**( *Float dist* )

Move the position left.

**moveNorth**( *Float dist* )

Move the position north.

**moveOffset**( *Vector dist* )

Move the position by the offset vector.

**moveRight**( *Float dist* )

Move the position right.

**moveSouth**( *Float dist* )

Move the position south.

**moveto**( *String vector\_or\_entity* )

Move to the specified vector or entity.

**moveUp**( *Float dist* )

Move the position up.

**moveWest**( *Float dist* )

Move the position west.

**next**

Goto the next waypoint.

**notshootable**

Makes the script slave not shootable

**openportal**

Open the area portal enclosed in this object

**physics\_off**

Turn physics off this script object on

**physics\_on**( *[ Integer no\_collide\_entity ]* )

Turn physics on this script object on  
If no\_collide\_entity is set to 1 then the script slave will not collide with other entities

**physics\_velocity**( *Vector impulseVector* )

Add a physical impulse to an object when it is being physically simulated

**rotateaxis**( *Integer axis, Float avelocity* )

Rotate about the specified axis at the specified angular velocity.

**rotateaxisdown**( *Integer axis, Float angle* )

Rotate the specified axis down by the specified amount.

**rotateaxisdownto**( *Integer axis, Float angle* )

Rotate the specified axis down to angle.

**rotateaxisup**( *Integer axis, Float angle* )

Rotate the specified axis up by the specified amount.

**rotateaxisupto**( *Integer axis, Float angle* )

Rotate the specified axis up to angle.

**rotatedownto**( *Vector direction* )

Rotate down to the specified direction.

**rotateto**( *Vector direction* )

Rotate to the specified direction.

**rotateupto**( *Vector direction* )

Rotate up to the specified direction.

**rotateX**( *Float avelocity* )

Rotate about the x axis at the specified angular velocity.

**rotateXdown**( *Float angle* )

Rotate the x down by the specified amount.

**rotateXdownto**( *Float angle* )

Rotate the x down to angle.

**rotateXup**( *Float angle* )

Rotate the x up by the specified amount.

**rotateXupto**( *Float angle* )

Rotate the x up to angle.

**rotateY**( *Float avelocity* )

Rotate about the y axis at the specified angular velocity.

**rotateYdown**( *Float angle* )

Rotate the y down by the specified amount.

**rotateYdownto**( *Float angle* )

Rotate the y down to angle.

**rotateYup**( *Float angle* )

Rotate the y up by the specified amount.

**rotateYupto**( *Float angle* )

Rotate the y up to angle.

**rotateZ**( *Float avelocity* )

Rotate about the z axis at the specified angular velocity.

**rotateZdown**( *Float angle* )

Rotate the z down by the specified amount.

**rotateZdownto**( *Float angle* )

Rotate the z down to angle.

**rotateZup**( *Float angle* )

Rotate the z up by the specified amount.

**rotateZupto**( *Float angle* )

Rotate the z up to angle.

**scriptslave\_followingpath**

Called every frame to actually follow the path

**scriptslave\_movedone**

Called when the script slave is done moving

**setdamage**( *Integer damage* )

Set the damage.

**setmeansofdeath**( *String means\_of\_death* )

Set the damage means of death.

**speed**( *Float speed* )

Sets the speed.

**stop**

Stop the script slave.

**threadmove**( *String label* )

Move the script slave and create thread which waits until finished.

**time**( *Float travel\_time* )

Sets the travel time.

**trigger**( *Entity ent* )

Trigger entities target.

**unbind**

unbind this entity.

**waitmove**

Move the script slave and wait until finished.

**ScriptThread ->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**abs**( *Float arg* )

Absolute value of int or float

**addobjective**( *Integer objective\_number, Integer status, [ String text ], [ Vector location ]* )

Adds/Changes an Objective

**aliascache**( *String alias, String realPath, [ String arg1 ], [ String arg2 ], [ String arg3 ], [ String arg4 ], [ String arg5 ], [ String arg6 ]* )

Create an alias to the specified path and cache the resource

**all\_ai\_off**

Turns all AI off.

**all\_ai\_on**

Turns all AI back on.

**angles\_pointat**( *Entity parent\_entity, Entity entity, Entity target\_entity* )

Returns the angles that points at the target\_entity given the base orientation of the parent\_entity and the position of the entity.

**angles\_toforward**( *Vector angles* )

Returns the forward vector of the specified angles

**angles\_toleft**( *Vector angles* )

Returns the left vector of the specified angles

**angles\_toup**( *Vector angles* )

Returns the up vector of the specified angles

**assert**( *Float value* )

Assert if value is 0.

**bool**( *Integer value* )

Casts value to a bool.

**bsptransition**( *String next\_map* )

Transitions to the next BSP. Keeps player data, and game data.

**cache**( *String resource\_name* )

Cache the specified resource.

**cam**( *String command, [ String arg1 ], [ String arg2 ], [ String arg3 ], [ String arg4 ], [ String arg5 ], [ String arg6 ]* )

Processes a camera command.

**centerprint**( *String stuffToPrint* )

prints the included message in the middle of all player's screens

**cinematic**

Turns on cinematic.

**clear\_objective\_pos**

Clears the position of the current objective, for when you don't have one

**clearfade**

Clear the fade from the screen

**clearletterbox**( *Float time* )

Clears letterbox mode.

**CreateListener**

Creates a Listener instance.

**cuecamera**( *Entity entity, [ Float switchTime ]* )

Cue the camera. If switchTime is specified, then the camera  
will switch over that length of time.

**cueplayer**( *[ Float switchTime ]* )

Go back to the normal camera. If switchTime is specified,  
then the camera will switch over that length of time.

**delaythrow**( *String label* )

Internal usage.

**drawhud**( *Integer value* )

Specfiy if hud is to be drawn

**earthquake**( *Float duration, Float magnitude, Float no\_rampup, Float no\_rampdown* )

Create an earthquake

**end**

End the thread.

**entity**( *Integer value* )

Casts value to an entity.

**error**( *String message, Integer level* )

Generate a script error with specified message and stack level

**exec**( *String script* )

Executes the specified script.

**exec**( *String script* )

Executes the specified script.

**fadein**( *Float time, Float red, Float green, Float blue, Float alpha, [ Integer mode ]* )

Sets up fadein in values.

**fadeout**( *Float time, Float red, Float green, Float blue, Float alpha, [ Integer mode ]* )

Sets up fadeout values.

**fadesound**( *Float time* )

fades the sound out over the given time.

**float**( *Integer value* )

Casts value to a float.

**forcemusic**( *String current, [ String fallback ]* )

Forces the current and fallback (optional) music moods.

**freezeplayer**

Freeze the player.

**getboundkey1**( *String keyname* )

return a string describing the key

**getboundkey2**( *String keyname* )

return a string describing the key

**getcvar**( *String name* )

getcvar

**goto**( *String label* )

Goes to the specified label.

**hidemenu**( *String name, [ Integer bForce ]* )

hide menu, with option to force it off

**hidemouse**

hide mouse cursor

**huddraw\_align**( *Integer index, String h\_align, String v\_align* )

Sets the alignment of a huddraw element. Specified with 'left', 'center', or 'right'

**huddraw\_alpha**( *Integer index, Float alpha* )

Sets the alpha of a huddraw element.

**huddraw\_color**( *Integer index, Float red, Float green, Float blue* )

Sets the color for a huddraw element

**huddraw\_font**( *Integer index, String fontname* )

Sets the font to use.

**huddraw\_rect**( *Integer index, Integer x, Integer y, Integer width, Integer height* )

Specifies the position of the upper left corner and size of a huddraw element

**huddraw\_shader**( *Integer index, String shader* )

Sets the shader to use for a particular huddraw element

**huddraw\_string**( *Integer index, String string* )

Sets a string to be displayed. Clears the shader value.

**huddraw\_virtualsize**( *Integer index, Integer virtual* )

Sets if the huddraw element should use virutal screen resolution for positioning and size

**int**( *Integer value* )

Casts value to an int.

**iprintln**( *String string* )

Prints a string.followed by a newline.

**iprintln\_noloc**( *String string* )

Prints a string.followed by a newline with no localization conversion.

**iprintlnbold**( *String string* )

Prints a string.followed by a newline in a bold/important way.

**iprintlnbold\_noloc**( *String string* )

Prints a string.followed by a newline in a bold/important way with no localization conversion.

**IsAlive**( *Entity ent* )

Returns true if the specified entity exists and has health > 0.

**killclass**( *String class\_name, [ Integer except ]* )

Kills everything in the specified class except for the specified entity (optional).

**killent**( *Integer ent\_num* )

Kill the specified entity.

**letterbox**( *Float time* )

Puts the game in letterbox mode.

**leveltransition**( *String next\_map* )

Transitions to the next Level. Statistics to Map Loading, does not keep player data or game data.

**loc\_convert\_string**( *String in* )

returns a localized version of the string.

**locprint**( *Integer xoffset, Integer yoffset, String stuffToPrint* )

prints the included message in the specified location of all player's screens

**map**( *String map\_name* )

Starts the specified map.

**missionfailed**

Makes the player fail their mission, level restarts.

**missiontransition**( *String next\_map* )

Transitions to the next Mission. Statistics to Main Menu, Next Level should be unlocked.

**mprint**( *String string* )

Prints a string.

**mprintln**( *String string* )

Prints a string.followed by a newline.

**music**( *String current, [ String fallback ]* )

Sets the current and fallback (optional) music moods.

**musicvolume**( *Float volume, Float fade\_time* )

Sets the volume and fade time of the music.

**noncinematic**

Turns off cinematic.

**pause**

Pauses the thread.

**popmenu**( *Integer index* )

pop menu

**print**( *String string* )

Prints a string.

**print3d**( *Vector origin, String string, [ String [strings] ]* )

prints a string in 3D space

**println**( *String string* )

Prints a string.followed by a newline.

**pushmenu**( *String name* )

push menu

**radiusdamage**( *Vector origin, Float damage, Float radius, Integer constant\_damage* )

radius damage at origin

**randomfloat**( *Float max* )

randomfloat

**randomint**( *Integer max* )

randomint

**releaseplayer**

Release the player.

**removeclass**( *String class\_name, [ Integer except ]* )

Removes everything in the specified class except for the specified entity (optional).

**removeent**( *Integer ent\_num* )

Removes the specified entity.

**restoremusicvolume**( *Float fade\_time* )

Restores the music volume to its previous value.

**restoresoundtrack**

Restores the soundtrack to the previous one.

**self**

self

**server**( *[ String arg1 ], [ String arg2 ], [ String arg3 ], [ String arg4 ], [ String arg5 ], [ String arg6 ]* )

Server only command.

**set\_objective\_pos**( *Vector pos* )

Sets the position in the world of the current objective

**setcurrentobjective**( *Integer objective\_number* )

Sets the specified objective as the current objective

**setcvar**( *String cvar\_name, String value* )

Sets the value of the specified cvar.

**setlightstyle**( *Integer lightstyleindex, String lightstyledata* )

Set up the lightstyle with lightstyleindex to the specified data

**showmenu**( *String name, [ Integer bForce ]* )

show menu, with option to force it on

**sighttrace**( *Vector start, Vector end, [ Integer pass\_entities ], [ Vector mins ], [ Vector maxs ]* )

Performs a trace line from the start to the end, returns 0 if something was hit and 1 otherwise

**soundtrack**( *String soundtrack\_name* )

Changes the soundtrack.

**spawn**( *String entityname, [ String keyname1 ], [ String value1 ], [ String keyname2 ], [ String value2 ], [ String keyname3 ], [ String value3 ], [ String keyname4 ], [ String value4 ]* )

Spawn the specified entity.

**spawn**( *String entityname, [ String keyname1 ], [ String value1 ], [ String keyname2 ], [ String value2 ], [ String keyname3 ], [ String value3 ], [ String keyname4 ], [ String value4 ]* )

Spawn the specified entity.

**string**( *Integer value* )

Casts value to a string.

**stuffcmd**( *[ String arg1 ], [ String arg2 ], [ String arg3 ], [ String arg4 ], [ String arg5 ], [ String arg6 ]* )

Server only command.

**teamwin**( *String axis\_or\_allies* )

Sets that the 'axis' or the 'allies' have won the map.

**thread**( *String label* )

Creates a thread starting at label.

**thread**( *String label* )

Creates a thread starting at label.

**throw**( *String label* )

Throws to the specified label.

**timeout**( *Float time* )

specifies script timeout time

**trace**( *Vector start, Vector end, [ Integer pass\_entities ], [ Vector mins ], [ Vector maxs ]* )

Performs a Trace Line from the start to the end, returns the end or the position it hit at

**trigger**( *String name* )

Trigger the specified target or entity.

**vector\_add**( *Vector vector1, Vector vector2* )

Returns vector1 + vector2

**vector\_closer**( *Vector vec\_a, Vector vec\_b, Vector vec\_c* )

returns 1 if the first vector is closer than the second vector to the third vector

**vector\_cross**( *Vector vector1, Vector vector2* )

Returns vector1 x vector2

**vector\_dot**( *Vector vector1, Vector vector2* )

Returns vector1 \* vector2

**vector\_length**( *Vector vector* )

Returns the length of the specified vector.

**vector\_normalize**( *Vector vector* )

Returns the normalized vector of the specified vector.

**vector\_scale**( *Vector vector1, Float scale\_factor* )

Returns vector1 \* scale\_factor

**vector\_subtract**( *Vector vector1, Vector vector2* )

Returns vector1 - vector2

**vector\_toangles**( *Vector vector1* )

Returns vector1 converted to angles.

**vector\_within**( *Vector position1, Vector position2, Float distance* )

returns 1 if the two points are <= distance apart, or 0 if they are greater than distance apart

**wait**( *Float wait\_time* )

Wait for the specified amount of time.

**waitframe**

Wait for one server frame.

**ScriptVariableList ->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**Sentient ->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**activatenewweapon**( *[ String handsurf ]* )

Activate the new weapon specified by useWeapon. handsurf allows specifying which hand to use for the player

**american**

Makes the sentient an American.

**ammo**( *String type, Integer amount* )

Gives the sentient some ammo.

**armor**( *String type, Integer amount* )

Gives the sentient some armor.

**blockend**

Is the end of the sentient's block.

**blockstart**

Is the start of the sentient's block.

**bloodmodel**( *String bloodModel* )

set the model to be used when showing blood

**charge**( *[ String hand ], [ String mode ]* )

Starts the charging of the weapon in the specified hand

**checkanims**

Check the animations in the .tik file versus the statefile

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**damagemult**( *Integer location, Float multiplier* )

Sets the damage multiplier for a particular body location

**deactivateweapon**( *String side* )

Deactivate the weapon in the specified hand.

**dontdropweapons**

Make the sentient not drop weapons

**dropitems**

drops inventory items

**fire**( *[ String hand ], [ String mode ]* )

Fires the weapon in the specified hand.

**german**

Makes the sentient a German.

**give**( *String name* )

Gives the sentient the targeted item.

**givedynitem**( *String model, String bonename* )

Pass the args to the item

**item**( *String type, Integer amount* )

Gives the sentient the specified amount of the specified item.

**jumpxy**( *Float forwardmove, Float sidemove, Float speed* )

Makes the sentient jump.

**maxgibs**( *Integer max\_number\_of\_gibs* )

Sets the maximum amount of generic gibs this sentient will spawn when hit.

**maxmouthangle**( *Float max\_mouth\_angle* )

Sets the max mouth angle.

**meleeattackend**

Is the end of the sentient's melee attack.

**meleeattackstart**

Is the start of the sentient's melee attack.

**noshadow**

Turns off the shadow for this sentient.

**onfire**

Called every frame when the sentient is on fire.

**pophelmet**

Pops a sentient's helmet off if he's got one

**putawayweapon**( *String whichHand* )

Put away or deactivate the current weapon, whichHand can be left, right or dual.

**releasefire**( *Float fireholdtime* )

Releases the attack in the time specified.

**reloadweapon**( *[ String hand ]* )

Reloads the weapon in the specified hand

**sethelmet**( *String tikifile, Float popspeed, Float dmgmult, String surfacename, String [optional* )

Gives the sentient a helmet and sets the needed info for it

**shadow**

Turns on the shadow for this sentient.

**spawnbloodygibs**( *[ Integer number\_of\_gibs ], [ Float scale ]* )

Spawns some bloody generic gibs.

**stopfire**( *[ String hand ]* )

Stops the firing of the weapon in the specified hand.

**stoponfire**

Stops the sentient from being on fire.

**stunend**

Is the end of the sentient's stun.

**stunstart**

Is the start of the sentient's stun.

**take**( *String item\_name* )

Takes away the specified item from the sentient.

**takeall**

Clears out the sentient's entire inventory.

**team**

returns 'german' or 'american'

**threatbias**

Gets the threat bias for this player / AI

**threatbias**( *Integer bias* )

Sets the threat bias for this player / AI

**threatbias**( *Integer bias* )

Sets the threat bias for this player / AI

**toggleitem**

Toggles the use of the player's item (first item if he has multiple)

**use**( *String name, [ Integer weapon\_hand ]* )

Use the specified weapon or item in the hand choosen (optional).

**uselast**

Activates the last active weapon

**useweaponclass**( *String weaponclass, [ Integer weapon\_hand ]* )

Use the weapon of the specified class in the hand choosen (optional).

**weapon**( *String weapon\_modelname* )

Gives the sentient the weapon specified.

**weaponcommand**( *String hand, [ String arg1 ], [ String arg2 ], [ String arg3 ], [ String arg4 ], [ String arg5 ], [ String arg6 ], [ String arg7 ]* )

Pass the args to the active weapon in the specified hand

**SimpleActor ->** [**Sentient**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Sentient) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**SimpleArchivedEntity ->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**SimpleEntity ->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**angle**

get the angles of the entity using just one value.  
Gets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**angles**( *Vector newAngles<0.00...360.00><0.00...360.00><0.00...360.00>* )

get the angles of the entity.

**angles**( *Vector newAngles<0.00...360.00><0.00...360.00><0.00...360.00>* )

set the angles of the entity to newAngles.

**angles**( *Vector newAngles<0.00...360.00><0.00...360.00><0.00...360.00>* )

set the angles of the entity to newAngles.

**centroid**

entity's centroid

**forwardvector**

get the forward vector of angles

**leftvector**

get the left vector of angles

**origin**

entity's origin

**origin**( *Vector newOrigin* )

Set the origin of the entity to newOrigin.

**origin**( *Vector newOrigin* )

Set the origin of the entity to newOrigin.

**rightvector**

get the right vector of angles

**target**( *String targetname\_to\_target* )

target another entity with targetname\_to\_target.

**target**( *String targetname\_to\_target* )

entity's target

**target**

entity's target

**targetname**

entity's targetname

**targetname**( *String targetName* )

set the targetname of the entity to targetName.

**targetname**( *String targetName* )

set the targetname of the entity to targetName.

**upvector**

get the up vector of angles

**SinkObject (*func\_sinkobject*) ->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**active**

make the SinkObject active, so that it will respond to players touching it.

**dampening**( *Float newDampening* )

dampening of SinkObject.

**delay**( *Float delay* )

Delay until SinkObject starts falling.

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**limit**( *Float newLimit* )

maximum displacement of the SinkObject.

**notactive**

make the SinkObject not active, so that it won't respond to players touching it.

**reset**

Reset the SinkObject right now.

**resetdelay**( *Float newResetDelay* )

Delay between when sinkobject starts resetting.

**resetsound**( *String newResetSound* )

Sound played when sinkobject is resetting.

**resetspeed**( *Float newResetspeed* )

Speed at which SinkObject resets itself, defaults to 0.002 \* speed.

**sinksound**( *String newSinkSound* )

Sound played when sinkobject is sinking.

**speed**( *Float speed* )

Speed at which SinkObject starts falling.

**SlidingDoor (*func\_door*) ->** [**Door**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Door) **->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**doclose**

Closes the door (special doors).

**doopen**( *Entity other* )

Opens the door (special doors).

**lip**( *Float lip* )

Sets the lip of the sliding door.

**speed**( *Float speed* )

Sets the speed of the sliding door.

**SoundManager ->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**addmusictrigger**

Add a new music trigger where the player is standing.

**addrandomspeaker**

Add a new sound where the player is standing.

**addreverbtrigger**

Add a new reverb trigger where the player is standing.

**addspeaker**

Add a new sound where the player is standing.

**delete**

Delete the current sound.

**globaltranslate**( *Vector translate\_amount* )

Translates all sounds and triggers by specified amount.

**hide**

Hides the sounds.

**moveplayer**

Move the player to the current sound position.

**next**

Go to the next sound.

**prev**

Go to the previous sound.

**previewreverb**

Test out the current reverb settings.

**replace**

Replace the current sound position with the player's.

**reset**

Resets the state of all sounds and triggers.

**resetreverb**

reset the reverb settings to a normal.

**save**

Saves the sounds.

**show**( *[ Entity path ]* )

Show all the sounds.

**switchfacet**

Switch the current facet that we are editing.

**updateinput**

Updates the current sound with user interface values.

**Spawn (*func\_spawn*) ->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**attackmode**( *Integer attackmode* )

Sets the attackmode for this spawn entity.

**doActivate**( *Entity activatingEntity* )

General trigger event for all entities

**modelname**( *String model\_name* )

Sets the model name for this spawn entity.

**pickup\_thread**( *String threadName* )

Sets the pickup thread for the spawned entity.

**spawnchance**( *Float spawn\_chance* )

Sets the chance that this spawned entity will spawn something when killed, if it is an actor.

**spawnitem**( *String spawn\_item\_name* )

Adds this named item to what will be spawned when this spawned entity is killed, if it is an actor.

**spawntarget**( *String spawntarget* )

Sets spawn target for this spawn entity.

**spawntargetname**( *String spawntargetname* )

Sets spawn target name for this spawn entity.

**SpawnArgs ->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**SpawnChain (*func\_spawnchain*) ->** [**Spawn**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Spawn) **->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**SpawnOutOfSight (*func\_spawnoutofsight*) ->** [**Spawn**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Spawn) **->** [**ScriptSlave**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#ScriptSlave) **->** [**Mover**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Mover) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**SplinePath (*info\_splinepath*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**fadetime**( *Float fadeTime* )

Sets the fadetime at this node.

**fov**( *Float cameraFOV* )

Sets the fov at this node.

**loop**( *String loop\_name* )

Sets the loop name.

**speed**( *Float speed* )

Sets the path speed.

**SplinePath\_create**

Creates the spline path from the target list.

**triggertarget**( *String target* )

Sets the trigger target.

**watch**( *String watchEntity* )

Sets the entity to watch at this node.

**StateScript ->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**TargetList ->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**Teleporter (*trigger\_teleport*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**TeleporterDestination (*func\_teleportdest*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**TempWaypoint ->** [**SimpleArchivedEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleArchivedEntity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**TestPlayerStart (*testplayerstart*) ->** [**PlayerStart**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#PlayerStart) **->** [**SimpleArchivedEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleArchivedEntity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**ThrowObject (*func\_throwobject*) ->** [**Object**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Object) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**pickup**( *Entity entity, String tag\_name* )

Picks up this throw object and attaches it to the entity.

**pickupoffset**( *Vector pickup\_offset* )

Sets the pickup\_offset.

**throw**( *Entity owner, Float speed, Entity targetent, [ Float grav ]* )

Throw this throw object.

**throwsound**( *String throw\_sound* )

Sets the sound to play when object is thrown.

**TossObject (*TossObject*) ->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**bouncesound**( *String sound* )

When bouncing, what sound to play on impact

**bouncesoundchance**( *Float chance<0.00...1.00>* )

When bouncing, the chance that the bounce sound will be played

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**stopped**

sent when entity has stopped bouncing for MOVETYPE\_TOSS.

**TouchAnim (*func\_touchanim*) ->** [**UseAnim**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#UseAnim) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**TouchField ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**TreeModel (*plant\_tree*) ->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**start**

Initializes the tree a little

**Trigger (*trigger\_multiple*) ->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**activatetrigger**( *Entity triggering\_entity* )

Activates all of the targets for this trigger.

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**cnt**( *Integer count* )

Set the amount of times this trigger can be triggered

**cone**( *Float newTriggerCone* )

Sets the cone in which directed triggers will trigger.

**delay**( *Float delay\_time* )

Set the delay time (time between triggering and firing) for this trigger

**doActivate**( *Entity activatingEntity* )

General trigger event for all entities

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**edgetriggered**( *Boolean newEdgeTriggered* )

If true, trigger will only trigger when object enters trigger, not when it is inside it.

**message**( *String message* )

Set a message to be displayed when this trigger is activated

**model**( *String modelName* )

set the model to modelName.

**multifaceted**( *Integer facetDirection* )

Make this trigger multifaceted. If facet is 1, than trigger is North/South oriented.  
If facet is 2 than trigger is East/West oriented. If facet is 3 than trigger is Up/Down oriented.

**noise**( *String sound* )

Set the sound to play when this trigger is activated

**nottriggerable**

Turn this trigger off

**setthread**( *String thread* )

Set the thread to execute when this trigger is activated

**sound**( *String sound* )

Set the sound to play when this trigger is activated

**triggerable**

Turn this trigger back on

**triggerthread**

Start the trigger thread.

**wait**( *Float wait\_time* )

Set the wait time (time bewteen triggerings) for this trigger

**TriggerAll (*trigger\_multipleall*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**doActivate**( *Entity activatingEntity* )

General trigger event for all entities

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**TriggerBox (*trigger\_box*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**maxs**( *Vector maxs* )

Sets the maximum bounds of the trigger box.

**mins**( *Vector mins* )

Sets the minimum bounds of the trigger box.

**TriggerByPushObject (*trigger\_pushobject*) ->** [**TriggerOnce**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#TriggerOnce) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**triggername**( *String targetname\_of\_object* )

If set, trigger will only respond to objects with specified name.

**TriggerCameraUse (*trigger\_camerause*) ->** [**TriggerUse**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#TriggerUse) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**doUse**( *Entity activatingEntity* )

sent to entity when it is used by another entity

**TriggerChangeLevel (*trigger\_changelevel*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**map**( *String map\_name* )

Sets the map to change to when triggered.

**spawnspot**( *String spawn\_spot* )

Sets the spawn spot to use.

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**TriggerClickItem (*trigger\_clickitem*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**model**( *String modelName* )

set the model to modelName.

**TriggerDamageTargets (*trigger\_damagetargets*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**activatetrigger**( *Entity triggering\_entity* )

Activates all of the targets for this trigger.

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**TriggerExit (*trigger\_exit*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**TriggerGivePowerup (*trigger\_givepowerup*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**oneshot**

Make this a one time trigger.

**powerupname**( *String powerup\_name* )

Specifies the powerup to give to the sentient.

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**TriggerHurt (*trigger\_hurt*) ->** [**TriggerUse**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#TriggerUse) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**damage**( *Integer damage* )

Sets the amount of damage to do.

**damagetype**( *String damageType* )

Sets the type of damage a TriggerHurt delivers.

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**TriggerMusic (*trigger\_music*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**altcurrent**( *String alternate\_current\_mood* )

Sets the alternate current mood to use when triggered.

**altfallback**( *String alterante\_fallback\_mood* )

Sets the alternate fallback mood to use when triggered.

**current**( *String current\_mood* )

Sets the current mood to use when triggered.

**fallback**( *String fallback\_mood* )

Sets the fallback mood to use when triggered.

**oneshot**

Make this a one time trigger.

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**triggereffectalt**( *Entity triggering\_entity* )

Send event to owner of trigger. This event is only triggered when using a trigger  
as a multi-faceted edge trigger.

**TriggerOnce (*trigger\_once*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**TriggerPlaySound (*play\_sound\_triggered*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**channel**( *Integer channel* )

Sets the sound channel to play on.

**min\_dist**( *Float min\_dist* )

Sets the minimum distance.

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**volume**( *Float volume* )

Sets the volume.

**TriggerPush (*trigger\_push*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**angle**( *Float newAngle* )

set the angles of the entity using just one value.  
Sets the yaw of the entity or an up and down  
direction if newAngle is [0-359] or -1 or -2

**speed**( *Float speed* )

Set the push speed of the TriggerPush

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**TriggerPushAny (*trigger\_pushany*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**speed**( *Float speed* )

Set the speed.

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**TriggerRelay (*trigger\_relay*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**TriggerReverb (*trigger\_music*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**altreverblevel**( *Float reverbLevel* )

Sets the reverb level to be used when triggered.

**altreverbtype**( *Integer reverbType* )

Sets the reverb type.

**oneshot**

Make this a one time trigger.

**reverblevel**( *Float reverbLevel* )

Sets the reverb level to be used when triggered.

**reverbtype**( *Integer reverbType* )

Sets the reverb type.

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**triggereffectalt**( *Entity triggering\_entity* )

Send event to owner of trigger. This event is only triggered when using a trigger  
as a multi-faceted edge trigger.

**TriggerSave (*trigger\_save*) ->** [**TriggerOnce**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#TriggerOnce) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**savename**( *String name* )

Sets the name which is appended to the world.message for this specific autosave

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**TriggerSecret (*trigger\_secret*) ->** [**TriggerOnce**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#TriggerOnce) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**TriggerSpeaker (*sound\_speaker*) ->** [**TriggerPlaySound**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#TriggerPlaySound) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**TriggerUse (*trigger\_use*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**doUse**( *Entity activatingEntity* )

sent to entity when it is used by another entity

**TriggerUseOnce (*trigger\_useonce*) ->** [**TriggerUse**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#TriggerUse) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**TriggerVehicle (*trigger\_vehicle*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**TurretGun ->** [**Weapon**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Weapon) **->** [**Item**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Item) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**burstFireSettings**( *Float mintime, Float maxtime, Float mindelay, Float maxdelay* )

Sets the settings for burst mode firing

**clearAimTarget**( *[ String target ]* )

Makes the turret aim at an entity

**doUse**( *Entity activatingEntity* )

sent to entity when it is used by another entity

**idleCheckOffset**( *Vector offset* )

Sets the offset to trace to for collision checking when idling

**item\_droptofloor**

Drops the item to the ground.

**maxYawOffset**( *Float maxoffset* )

Sets the max yaw offset from the turrets central facing direction

**pitchCaps**( *Vector caps* )

Sets the pitch caps for the turret. First number is upward cap, second is downward cap, and the third just makes it nice little vector

**setAimOffset**( *Vector offset* )

Makes the turret aim with specified offset

**setAimTarget**( *[ String target ]* )

Makes the turret aim at an entity

**setPlayerUsable**( *Integer state* )

Sets wether the turret can be used by players. 0 means no, 1 means yes.

**setthread**( *String value* )

Sets the name of the thread called when a player uses the turret

**shoot**( *[ String mode ]* )

Shoot the weapon

**startFiring**

Makes the turret start shooting

**stopFiring**

Makes the turret stop shooting

**turnSpeed**( *Float speed* )

Sets the turret's turn speed

**userdistance**( *Float dist* )

Sets the distance the user should be placed at while using this turret

**viewangles**

get the angles of the entity.

**viewangles**( *Vector newAngles* )

set the view angles of the entity to newAngles.

**viewjitter**( *Float amount* )

Sets the amount that the owner's view should be jittered when fired

**viewOffset**( *Vector offset* )

Sets the view offset to use for the turret

**yawCenter**( *Float yaw* )

Sets the yaw for the center of the turret's turning arc

**UseAnim (*func\_useanim*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**anim**( *String animName* )

set the animation to use for player.

**camera**( *String cameraPosition* )

set the camera to use when in this animation.  
topdown, behind, front, side, behind\_fixed, side\_left, side\_right

**count**( *Integer newCount* )

Sets how many times the UseAnim can be triggered.

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**key**( *String keyName* )

set the key needed to make this UseAnim function.

**num\_loops**( *Integer loopCount* )

set the number of times to loop an animation per use.

**state**( *String stateName* )

set the state to use for the player.

**triggertarget**( *String targetname* )

Sets what should be triggered, when this UseAnim is triggered.

**UseAnimDestination (*func\_useanimdest*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**anim**( *String animName* )

set the animation to use for player.

**num\_loops**( *Integer loopCount* )

set the number of times to loop an animation per use.

**state**( *String stateName* )

set the state to use for the player.

**UseObject (*func\_useobject*) ->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**activate**

Allow the useobject to be used.

**cone**( *Float newCone* )

Sets the cone in angles of where the Useobject can be used.

**count**( *Integer newCount* )

Sets how many times the UseObject can be triggered.

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**damage\_type**( *String newDamageType* )

Sets what kind of damage is needed to activate the trigger.

**deactivate**

Do not allow the useobject to be used.

**offset**( *Vector newOffset* )

Sets the offset to use for this UseObject.

**reset\_time**( *Float newResetTime* )

Sets the time it takes for the UseObject to reset itself.

**state**( *String newState* )

Sets the state to use for this UseObject.

**state\_backwards**( *String newState* )

Sets the backward state to use for this UseObject.

**triggertarget**( *String targetname* )

Sets what should be triggered, when this UseObject is triggered.

**usematerial**( *String nameOfUseMaterial* )

the name of the material that glows when active.

**yaw\_offset**( *Float newYawOffset* )

Sets the yaw offset to use for this UseObject.

**Vehicle (*script\_vehicle*) ->** [**VehicleBase**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#VehicleBase) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**AnimationSet**( *String animset* )

Sets the Animation Set to use.

**AttachDriverSlot**( *Integer slot, Entity entity* )

Attaches an entity to the specified slot.

**AttachPassengerSlot**( *Integer slot, Entity entity* )

Attaches an entity to the specified slot.

**AttachTurretSlot**( *Integer slot, Entity entity* )

Attaches an entity to the specified slot.

**back\_mass**( *Float weight* )

Sets the mass of the back of the vehicle

**canjump**( *Boolean jumpable* )

Sets whether or not the vehicle can jump

**collisionent**

Gets the Collision Entity

**collisionent**( *Entity entity* )

Gets the Collision Entity

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**DetachDriverSlot**( *Integer slot, [ Vector exit\_position ]* )

Detaches an entity to the specified slot.

**DetachPassengerSlot**( *Integer slot, [ Vector exit\_position ]* )

Detaches an entity to the specified slot.

**DetachTurretSlot**( *Integer slot, [ Vector exit\_position ]* )

Detaches an entity to the specified slot.

**doBlocked**( *Entity obstacle* )

sent to entity when blocked.

**doTouch**( *Entity touchingEntity* )

sent to entity when touched.

**doUse**( *Entity activatingEntity* )

sent to entity when it is used by another entity

**drivable**

Make the vehicle drivable

**drive**( *Vector position, Float speed, Float acceleration, Float reach\_distance, Float look\_ahead, [ Vector alternate\_position ]* )

Makes the vehicle drive to position with speed and acceleration until reached\_distance close to position

**driveNoWait**( *Vector position, Float speed, Float acceleration, Float reach\_distance* )

Makes the vehicle drive to position with speed and acceleration until reached\_distance close to position, thread doesn't wait

**explosionmodel**( *String model* )

Sets the TIKI to call when the vehicle dies.

**front\_mass**( *Float weight* )

Sets the mass of the front of the vehicle

**fullstop**

Make the Vehicle Stop Moving... Completely!

**lock**

Sets the vehicle to be locked

**lockmovement**

The Vehicle cannot move.

**model**( *String modelName* )

set the model to modelName.

**modifydrive**( *Float desired\_speed, Float acceleration, Float look\_ahead* )

Modifys the parameters of the current drive.

**nextdrive**( *Entity next\_path* )

appends the specified path to the current path

**QueryDriverSlotAngles**( *Integer slot* )

Returns the angles of the specified slot on the vehicle.

**QueryDriverSlotEntity**( *Integer slot* )

Returns an entity at the specified slot.

**QueryDriverSlotPosition**( *Integer slot* )

Returns the position of the specified slot on the vehicle.

**QueryDriverSlotStatus**( *Integer slot* )

Returns the status of the specified slot on the vehicle.

**QueryFreeDriverSlot**

Returns a number that represents the first free driver slot on the vehicle.

**QueryFreePassengerSlot**

Returns a number that represents the first free passenger slot on the vehicle.

**QueryFreeTurretSlot**

Returns a number that represents the first free turret slot on the vehicle.

**QueryPassengerSlotEntity**( *Integer slot* )

Returns an entity at the specified slot.

**QueryPassengerSlotPosition**( *Integer slot* )

Returns the position of the specified slot on the vehicle.

**QueryPassengerSlotStatus**( *Integer slot* )

Returns the status of the specified slot on the vehicle.

**QueryTurretSlotEntity**( *Integer slot* )

Returns an entity at the specified slot.

**QueryTurretSlotPosition**( *Integer slot* )

Returns the position of the specified slot on the vehicle.

**QueryTurretSlotStatus**( *Integer slot* )

Returns the status of the specified slot on the vehicle.

**removeondeath**( *Integer removeondeath* )

If set to a non-zero value, vehicles will not be removed when they die

**seatanglesoffset**( *Vector angles* )

Set the angles offset of the seat

**seatoffset**( *Vector offset* )

Set the offset of the seat

**setcollisionentity**( *Entity entity* )

Sets the Collision Entity.

**setsoundparameters**( *Float min\_speed, Float min\_pitch, Float max\_speed, Float max\_pitch* )

Sets the Sound parameters for this vehicle

**setvolumeparameters**( *Float min\_speed, Float min\_volume, Float max\_speed, Float max\_volume* )

Sets the Volume parameters for this vehicle

**setweapon**( *String weaponname* )

Set the weapon for the vehicle

**showweapon**

Set the weapon to be show in the view

**skidding**( *Integer on\_off* )

Makes the vehicle skid around corners.

**SoundSet**( *String soundset* )

Sets the Sound Set to use.

**spawnturret**( *Integer slot, String tikifile* )

Spawns a turret with the specified model and connects it to the specified slot

**start**

Initialize the vehicle.

**steerinplace**

Set the vehicle to turn in place

**stop**

Make the Vehicle Stop Moving... FULL BREAKS!

**stopatend**

Makes the vehicle slow down to a complete stop at the end of the path.

**turnrate**( *Float rate* )

Set the turning rate of the vehicle

**undrivable**

Make the vehicle undrivable

**unlock**

Sets the vehicle to be unlocked

**unlockmovement**

The Vehicle can move again.

**vehicleanim**( *String anim\_name, [ Float weight ]* )

Sets an animation to use in the LD Animation slot.   
Weight defaults to 1.0

**vehiclebouncy**( *Float bouncycoef* )

Sets the Bouncy Coefficient for the shocks.

**vehicledestroyed**

Driver is dead

**vehicledrag**( *Float size* )

Sets the Drag Factor

**vehicleinit**

Initialized the Vehicle as the specified file

**vehiclemass**( *Float weight* )

Sets the mass of the vehicle (backmass = frontmass = mass/2)

**vehicleradius**( *Float size* )

Sets the radius of the wheels

**vehicleRoll**( *Float min, Float max, Float coef* )

Sets the Roll min and max and the acceleration coefficient for the shocks.

**vehiclerollingresistance**( *Float size* )

Sets the radius of the wheels

**vehiclespeed**( *Float speed* )

Set the speed of the vehicle

**vehiclespringy**( *Float springycoef* )

Sets the Springy Coefficient for the shocks.

**vehicletread**( *Float size* )

Sets the size of the wheels

**VehicleWheelCorners**( *Vector size, Vector offset* )

Sets the wheel trace corners.

**vehicleYaw**( *Float min, Float max, Float coef* )

Sets the Yaw min and max and the acceleration coefficient for the shocks.

**vehicleZ**( *Float min, Float max, Float coef* )

Sets the Z min and max and the acceleration coefficient for the shocks.

**VehicleBase ->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**VehicleCollisionEntity ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**VehicleHalfTrack (*VehicleHalfTrack*) ->** [**DrivableVehicle**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#DrivableVehicle) **->** [**Vehicle**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Vehicle) **->** [**VehicleBase**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#VehicleBase) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**killed**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

event which is sent to an entity once it as been killed  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**VehiclePoint (*info\_vehiclepoint*) ->** [**Waypoint**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Waypoint) **->** [**SimpleArchivedEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleArchivedEntity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**spawnflags**( *Integer spawn\_flags* )

Sets the spawn flags.

**VehicleSoundEntity ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**vehiclesoudnentity\_updatetraces**

Updates the traces of a Vehicle Sound Entity

**vehiclesoundentity\_postspawn**

PostSpawn of a Vehicle Sound Entity

**VehicleTank (*VehicleTank*) ->** [**DrivableVehicle**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#DrivableVehicle) **->** [**Vehicle**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Vehicle) **->** [**VehicleBase**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#VehicleBase) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**VehicleTurretGun ->** [**TurretGun**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#TurretGun) **->** [**Weapon**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Weapon) **->** [**Item**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Item) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**collisionent**

Gets the Collision Entity

**collisionent**( *Entity entity* )

Sets the Collision Entity

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**doUse**( *Entity activatingEntity* )

sent to entity when it is used by another entity

**item\_droptofloor**

Drops the item to the ground.

**killed**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

event which is sent to an entity once it as been killed  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**lock**

The Turret can not be used.

**removeondeath**( *Integer removeondeath* )

If set to a non-zero value, vehicles will not be removed when they die

**setbaseentity**( *Entity base\_entity* )

Sets the base entity to take its orientation from.

**setcollisionentity**( *Entity entity* )

Sets the Collision Entity.

**SoundSet**( *String soundset* )

Sets the Sound Set to use.

**turnSpeed**( *Float speed* )

Sets the turret's turn speed

**unlock**

The Turret Can be used.

**VehicleWheelsX2 (*VehicleWheelsX2*) ->** [**DrivableVehicle**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#DrivableVehicle) **->** [**Vehicle**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Vehicle) **->** [**VehicleBase**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#VehicleBase) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**killed**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

event which is sent to an entity once it as been killed  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**VehicleWheelsX4 (*VehicleWheelsX4*) ->** [**DrivableVehicle**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#DrivableVehicle) **->** [**Vehicle**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Vehicle) **->** [**VehicleBase**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#VehicleBase) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**killed**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

event which is sent to an entity once it as been killed  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**ViewJitter (*func\_viewjitter*) ->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**donedeath**

Makes the view jitter only happen once

**duration**( *Float time* )

Sets the length of time it should last. 0 will be instantanious

**edgeeffect**( *Float fraction* )

Sets the fraction of the jitter to apply at the max radius

**jitteramount**( *Vector jitterangles* )

Sets the jitter angles to apply to the player

**radius**( *Float radius* )

Sets the max radius of the view jitter. 0 affects all

**timedecay**( *Vector decayrate* )

Sets jitter decay per second

**triggereffect**( *Entity triggering\_entity* )

Send event to owner of trigger.

**ViewMaster ->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**viewangles**( *Float pitch<0.00...360.00>, Float yaw<0.00...360.00>, Float roll<0.00...360.00>* )

Set the angles of the current viewthing

**viewanimate**

Cycle through the animations modes of the current viewthing  
No Animation  
Animation with no motion  
Animation with looping motion  
Animation with motion

**viewattach**( *String tagname, String model* )

Attach a model the the specified tagname

**viewdelete**

Delete the current viewthing

**viewdeleteall**

Delete all viewthings

**viewdetach**

Detach the current viewthing from its parent

**viewdetachall**

Detach all the models attached to the current viewthing

**viewmodel**( *String viewthingModel* )

Set the model of the current viewthing

**viewnext**

Advance to the next frame of animation of the current viewthing

**viewnextanim**

Advance to the next animation of the current viewthing

**vieworigin**( *Float x, Float y, Float z* )

Set the origin of the current viewthing

**viewpitch**( *Float pitch* )

Set the pitch of the current viewthing

**viewprev**

Advance to the previous frame of animation of the current viewthing

**viewprevanim**

Advance to the previous animation of the current viewthing

**viewroll**( *Float roll* )

Set the roll of the current viewthing

**viewscale**( *Float scale* )

Set the scale of the current viewthing

**viewscaledown**

Decrease the scale of the current viewthing

**viewscaleup**

Increase the scale of the current viewthing

**viewscrub**( *Float animScrub* )

Set the animation time based off a floating point value

**viewsetanim**( *Float animNum* )

Set the animation absolutely based off a floating point value

**viewsetanim2**( *Float animNum2* )

Set the animation absolutely based off a floating point value

**viewsetanimslot**( *Integer animSlot* )

Set current animation slot that sliders apply to

**viewspawn**( *String model* )

Create a viewthing with the specified model

**viewthingnext**

Change the active viewthing to the next viewthing

**viewthingprev**

Change the active viewthing to the previous viewthing

**viewyaw**( *Float yaw* )

Set the yaw of the current viewthing

**Viewthing (*viewthing*) ->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**exec**( *String script* )

Don't execute the specified script for ViewThing.

**viewangles**( *Float pitch<0.00...360.00>, Float yaw<0.00...360.00>, Float roll<0.00...360.00>* )

Set the angles of the current viewthing

**viewanimate**

Cycle through the animations modes of the current viewthing  
No Animation  
Animation with no motion  
Animation with looping motion  
Animation with motion

**viewattach**( *String tagname, String model* )

Attach a model the the specified tagname

**viewdelete**

Delete the current viewthing

**viewdetach**

Detach the current viewthing from its parent

**viewdetachall**

Detach all the models attached to the current viewthing

**viewlastframe**

Called when the view things last animation frame is displayed.

**viewmodel**( *String viewthingModel* )

Set the model of the current viewthing

**viewnext**

Advance to the next frame of animation of the current viewthing

**viewnextanim**

Advance to the next animation of the current viewthing

**vieworigin**( *Float x, Float y, Float z* )

Set the origin of the current viewthing

**viewpitch**( *Float pitch* )

Set the pitch of the current viewthing

**viewprev**

Advance to the previous frame of animation of the current viewthing

**viewprevanim**

Advance to the previous animation of the current viewthing

**viewroll**( *Float roll* )

Set the roll of the current viewthing

**viewsavesurfaces**

Called after the model is spawned to save off the models original surfaces.

**viewscale**( *Float scale* )

Set the scale of the current viewthing

**viewscaledown**

Decrease the scale of the current viewthing

**viewscaleup**

Increase the scale of the current viewthing

**viewscrub**( *Float animScrub* )

Set the animation time based off a floating point value

**viewsetanim**( *Float animNum* )

Set the animation absolutely based off a floating point value

**viewsetanim2**( *Float animNum2* )

Set the animation absolutely based off a floating point value

**viewsetanimslot**( *Integer animSlot* )

Set current animation slot that sliders apply to

**viewthing\_think**

Called every frame to process the view thing.

**viewyaw**( *Float yaw* )

Set the yaw of the current viewthing

**Waypoint (*info\_waypoint*) ->** [**SimpleArchivedEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleArchivedEntity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**Weapon ->** [**Item**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Item) **->** [**Trigger**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Trigger) **->** [**Animate**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Animate) **->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**ai\_event**( *[ String type ], [ Float radius ]* )

Let the AI know that this entity made a sound,  
type is a string specifying what type of sound it is.  
radius determines how far the sound reaches.

**airange**( *String airange* )

Set the range of this gun for the ai: short, medium, long, sniper

**ammo\_in\_clip**( *Integer ammoInClip* )

Set the amount of ammo in the clip

**ammopickupsound**( *String name* )

sets the weapon's ammo pickup sound alias

**ammorequired**( *Integer amount* )

Set the amount of ammo this weapon requires to fire

**ammotype**( *String name* )

Set the type of ammo this weapon uses

**anim**( *String animName* )

Exec anim commands on server or client.

**attachtohand**( *String hand* )

Attaches an active weapon to the specified hand

**autoaim**

Turn on auto aiming for the weapon

**autoputaway**( *Boolean bool* )

Set the weapon to be automatically put away when out of ammo

**bulletcount**( *Float bulletCount* )

Set the number of bullets this weapon shoots when fired

**bulletdamage**( *Float bulletDamage* )

Set the damage that the bullet causes

**bulletknockback**( *Float bulletKnockback* )

Set the knockback that the bullet causes

**bulletrange**( *Float bulletRange* )

Set the range of the bullets

**bulletspread**( *Float bulletSpreadX, Float bulletSpreadY, [ Float bulletSpreadXmax ], [ Float bulletSpreadYmax ]* )

Set the min & optional max spread of the bullet in the x and y axis

**cantpartialreload**

Prevents the weapon from being reloaded part way through a clip

**clip\_add**( *Integer ammoAmount* )

Add to the weapons ammo clip with ammo from its owner

**clip\_empty**

Empties the weapon's clip of ammo, returning it to the owner

**clip\_fill**

Fills the weapons ammo clip with ammo from its owner

**clipsize**( *Integer ammoClipSize* )

Set the amount of rounds a clip of the weapon holds

**crosshair**( *Boolean bool* )

Turn on/off the crosshair for this weapon

**dmammorequired**( *Integer amount* )

Set the amount of ammo this weapon requires to fire for DM

**dmbulletcount**( *Float bulletCount* )

Set the number of bullets this weapon shoots when fired for DM

**dmbulletdamage**( *Float bulletDamage* )

Set the damage that the bullet causes for DM

**dmbulletrange**( *Float bulletRange* )

Set the range of the bullets

**dmbulletspread**( *Float bulletSpreadX, Float bulletSpreadY, [ Float bulletSpreadXmax ], [ Float bulletSpreadYmax ]* )

Set the min & optional max spread of the bullet in the x and y axis

**dmcantpartialreload**

Prevents the weapon from being reloaded part way through a clip for DM

**dmcrosshair**( *Boolean bool* )

Turn on/off the crosshair for this weapon

**dmfiredelay**( *Float fFireDelay* )

Set the minimum time between shots from the weapon for DM

**dmfirespreadmult**( *Float scaleadd, Float falloff, Float cap* )

Sets a time decayed multiplyer to spread when the weapon is fired

**dmmovementspeed**( *Float speedmult* )

Alters the movement speed of the player when he has the weapon out

**dmprojectile**( *String projectileModel* )

Set the model of the projectile that this weapon fires in DM

**dmstartammo**( *Integer amount* )

Set the starting ammo of this weapon

**dmzoomspreadmult**( *Float scale* )

Sets the spread multiplyer for when using the zoom on a zooming weapon

**donefiring**

Signals the end of the fire animation

**donereloading**

Signals the end of the reload animation

**fallingangleadjust**

Adjusts the weapons angles as it falls to the ground

**firedelay**( *Float fFireDelay* )

Set the minimum time between shots from the weapon

**firespreadmult**( *Float scaleadd, Float falloff, Float cap, Float maxtime* )

Sets a time decayed multiplyer to spread when the weapon is fired

**firetype**( *String firingType* )

Set the firing type of the weapon (projectile or bullet)

**holsterangles**( *Vector angles* )

Set the angles of this weapon when it is holstered

**holsteroffset**( *Vector offset* )

Set the positional offset when it is holstered

**holsterscale**( *Float scale* )

Set the scale of the weapon when it's attached to the holster

**holstertag**( *String tagname* )

Set the name of the tag to attach this to when the weapon is holstered.

**idle**

Puts the weapon into an idle state

**idleinit**

Puts the weapon into an idle state and clears all the anim slots

**item\_pickup**( *Entity item* )

Pickup the specified item.

**loopfire**

Makes the weapon fire by looping the fire animation.

**mainattachtotag**( *String tagname* )

Set the name of the tag to attach this to it's owner when being used.

**makenoise**( *[ Float noise\_radius ], [ Boolean force ]* )

Makes the weapon make noise that actors can hear.

**maxchargetime**( *Integer time* )

Set the maximum time the weapon may be charged up

**maxrange**( *Float maxRange* )

Set the maximum range of a weapon so the AI knows how to use it

**meansofdeath**( *String meansOfDeath* )

Set the meansOfDeath of the weapon.

**minchargetime**( *Integer time* )

Set the minimum time the weapon must be charged up

**minrange**( *Float minRange* )

Set the minimum range of a weapon so the AI knows how to use it

**movementspeed**( *Float speedmult* )

Alters the movement speed of the player when he has the weapon out

**noammosound**( *String name* )

sets the weapon's dry fire sound alias

**notdroppable**

Makes a weapon not droppable

**offhandattachtotag**( *String tagname* )

Set the name of the tag to attach this to it's owner's off hand.

**projectile**( *String projectileModel* )

Set the model of the projectile that this weapon fires

**quiet**

Makes the weapon make no noise.

**range**( *Float range* )

Set the range of the weapon

**rank**( *Integer iOrder,, Integer iRank* )

Set the order value and power ranking for the weapon

**ready**

Signals the end of the ready animation so the weapon can be used

**secondary**( *[ String arg1 ], [ String arg2 ], [ String arg3 ], [ String arg4 ], [ String arg5 ], [ String arg6 ], [ String arg7 ], [ String arg8 ]* )

Set the secondary mode of the weapon, by passing commands through

**semiauto**

Sets the weapon to fire semi-auto

**setaimanim**( *String aimAnimation, Integer aimFrame* )

Set the aim animation and frame for when a weapon fires

**shareclip**( *Integer shareClip* )

Sets the weapon to share the same clip between all fire modes

**shoot**( *[ String mode ]* )

Shoot the weapon

**startammo**( *Integer amount* )

Set the starting ammo of this weapon

**startingammotoowner**

Internal event used to give ammo to the owner of the weapon

**tracerfrequency**( *Integer frequenct* )

Set the frequency of making tracers

**usenoammo**( *Boolean bool* )

Set the weapon to be able to be used when it's out of ammo

**viewkick**( *Float pitchmin, Float pitchmax, [ Float yawmin ], [ Float yawmax ]* )

Adds kick to the view of the owner when fired.

**weapongroup**( *String weapon\_group* )

Sets the weapon group, a set of animations for actor animations scripts to use

**weapontype**( *String weapon\_type* )

Sets the weapon type

**worldhitspawn**( *String modelname* )

Set a model to be spawned when the weapon strikes the world.

**zoom**( *Integer zoomfov, [ Integer autozoom ]* )

Sets fov to zoom to on a secondary fire

**zoomspreadmult**( *Float scale* )

Sets the spread multiplyer for when using the zoom on a zooming weapon

**WindowObject (*func\_window*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**damage**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

general damage event used by all entities  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**debristype**( *Integer type* )

Sets the debris type of the Window

**killed**( *Entity attacker, Integer damage, Entity inflictor, Vector position, Vector direction, Vector normal, Integer knockback, Integer damageflags, Integer meansofdeath, Integer location* )

event which is sent to an entity once it as been killed  
  
Location values:  
-1 General  
0 Pelvis  
1 Lower Torso  
2 Mid Torso  
3 Upper Torso  
4 Neck  
5 Head  
6 RUpperArm  
7 RForearm  
8 RHand  
9 LUpperArm  
10 LForearm  
11 LHand  
12 RThigh  
13 RCalf  
14 RFoot  
15 LThigh  
16 LCalf  
17 LFoot

**World (*worldspawn*) ->** [**Entity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Entity) **->** [**SimpleEntity**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#SimpleEntity) **->** [**Listener**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Listener) **->** [**Class**](http://gronnevik.se/rjukan/uploads/Main/g_allclasses.html#Class)

**ai\_visiondistance**( *Float vision\_distance* )

Sets the default AI Vision Distance

**farplane**

Get the distance of the far clipping plane

**farplane**( *Float farplaneDistance* )

Set the distance of the far clipping plane

**farplane**( *Float farplaneDistance* )

Set the distance of the far clipping plane

**farplane\_color**( *Vector farplaneColor* )

Set the color of the far clipping plane fog

**farplane\_color**

Get the color of the far clipping plane fog

**farplane\_color**( *Vector farplaneColor* )

Set the color of the far clipping plane fog

**farplane\_cull**( *Boolean farplaneCull* )

Whether or not the far clipping plane should cull things out of the world

**gravity**( *Float worldGravity* )

Set the gravity for the whole world.

**lavaalpha**( *Float lavaAlpha* )

Set the alpha of lava screen blend

**lavacolor**( *Vector lavaColor* )

Set the color of lava screen blend

**message**( *String worldMessage* )

Set a message for the world

**nextmap**( *String nextMap* )

Set the next map to change to

**northyaw**( *Float yaw* )

Sets the yaw direction that is considered to be north

**numarenas**( *Integer numarenas* )

Set the number of arenas in the world

**skyalpha**( *Float newAlphaForPortalSky* )

Set the alpha on the sky

**skyportal**( *Boolean newSkyPortalState* )

Whether or not to use the sky portal at all

**soundtrack**( *String MusicFile* )

Set music soundtrack for this level.

**wateralpha**( *Float waterAlpha* )

Set the alpha of the water screen blend

**watercolor**( *Vector waterColor* )

Set the watercolor screen blend

**154 Game Module Classes.  
1587 Game Module Events.**