

Half-Life 1 Game Events

From AMWiki

Contents

- 1 Introduction
- 2 ADStop
- 3 AllowSpec
- 4 AmmoPickup
- 5 AmmoX
- 6 ArmorType
- 7 BarTime
- 8 BarTime2
- 9 Battery
- 10 BlinkAcct
- 11 BombDrop
- 12 BombPickup
- 13 BotProgress
- 14 BotVoice
- 15 Brass
- 16 BuyClose
- 17 ClCorpse
- 18 Crosshair
- 19 CurWeapon
- 20 CZCareer
- 21 CZCareerHUD
- 22 Damage
- 23 DeathMsg
- 24 FlashBat
- 25 Flashlight
- 26 Fog
- 27 ForceCam
- 28 GameMode
- 29 GameTitle
- 30 Geiger
- 31 Health
- 32 HideWeapon
- 33 HLTV
- 34 HostageK
- 35 HostagePos
- 36 HudText
- 37 HudTextArgs
- 38 HudTextPro
- 39 InitHUD
- 40 ItemPickup
- 41 ItemStatus

- 42 Location
- 43 Money
- 44 MOTD
- 45 NVGToggle
- 46 Radar
- 47 ReceiveW
- 48 ReloadSound
- 49 ReqState
- 50 ResetHUD
- 51 RoundTime
- 52 SayText
- 53 Scenario
- 54 ScoreAttrib
- 55 ScoreInfo
- 56 ScreenFade
- 57 ScreenShake
- 58 SendAudio
- 59 ServerName
- 60 SetFOV
- 61 ShadowIdx
- 62 ShowMenu
- 63 ShowTimer
- 64 SpecHealth
- 65 SpecHealth2
- 66 Spectator
- 67 StatusIcon
- 68 StatusValue
- 69 StatusText
- 70 TaskTime
- 71 TeamInfo
- 72 TeamScore
- 73 TextMsg
- 74 Train
- 75 TutorClose
- 76 TutorLine
- 77 TutorState
- 78 TutorText
- 79 ViewMode
- 80 VGUIMenu
- 81 VoiceMask
- 82 WeaponList
- 83 WeapPickup

Introduction

In AMX Mod X you are able to hook in-game messages/events with `register_message` (http://www.amxmodx.org/funcwiki.php?search=register_message&go=search) / `register_event` (<http://www.amxmodx.org/funcwiki.php?go=func&id=26>) . Here are the list of messages/events and their arguments which can be read with `get_msg_arg_*` (http://www.amxmodx.org/funcwiki.php?search=get_msg_arg_&go=search) / `read_data` (<http://www.amxmodx.org/funcwiki.php?go=func&id=132>) .

- See the Advanced Scripting article for more on events and messages.
- See `messages.inc` and `message_const.inc` from `amxmodx/scriptong/include` folder or Messaging functions (<http://www.amxmodx.org/funcwiki.php?go=inc&id=47>) and Message constants (<http://www.amxmodx.org/funcwiki.php?go=inc&id=48>) for full messages control including blocking, argument alteration and much more.

ADStop

Note: No Information available for this message.

Note: This message has no arguments.

Name:	ADStop
Structure:	.

AllowSpec

Changes whether or not "SPECTATE" appears on the change team menu. Called whenever the `allow_spectators` CVar is changed, with its new value sent as the byte.

Note: This changes how the change team menu appears, but spectating functionality is based off of the actual CVar value.

Name:	AllowSpec		
Structure:	<table><tr><td>byte</td><td>Allowed</td></tr></table>	byte	Allowed
byte	Allowed		

AmmoPickup

Temporarily draws HUD ammo amount and corresponding ammo HUD icon in the middle of the right side of the screen.

Note: Draw time is depend on the `hud_drawhistory_time` client CVar value.

Note: See CS Weapons Information for more information.

Name:	AmmoPickup				
Structure:	<table><tr><td>byte</td><td>AmmoID</td></tr><tr><td>byte</td><td>Ammount</td></tr></table>	byte	AmmoID	byte	Ammount
byte	AmmoID				
byte	Ammount				

AmmoX

Updates the green bar indicator in the HUD weapons list. Also updates HUD backpack ammo number in the lower right corner of the screen in case the given ammo type are compatible with the current weapon.

Note: See CS Weapons Information for more information.

Name:	AmmoX				
Structure:	<table><tr><td>byte</td><td>AmmoID</td></tr><tr><td>byte</td><td>Ammount</td></tr></table>	byte	AmmoID	byte	Ammount
byte	AmmoID				
byte	Ammount				

ArmorType

Draws/removes helmet HUD icon. If flag is set to 1 the helmet HUD icon will be drawn (located right on the armor icon).

Name:	ArmorType		
Structure:	<table><tr><td>byte</td><td>Flag</td></tr></table>	byte	Flag
byte	Flag		

BarTime

Draws a HUD progress bar which is filled from 0% to 100% for the time Duration seconds. Once the bar is fully filled it will be removed from the screen automatically.

Note: Set Duration to 0 to hide the bar.

Name:	BarTime		
Structure:	<table><tr><td>short</td><td>Duration</td></tr></table>	short	Duration
short	Duration		

BarTime2

The same as BarTime but StartPercents is specify how many per cents of the bar are (already) filled.

Note: Display time can be calculated with this formula: $(1 - (StartPercents / 100)) / Duration$

Name: Structure:	BarTime2	
	short short	Duration StartPercents

Battery

Updates the icon and the number of armor on the HUD.

Name: Structure:	Battery	
	short	Armor

BlinkAcct

Makes a player's money display flash rapidly, until it flashes a total of BlinkAmt times.

Name: Structure:	BlinkAcct	
	byte	BlinkAmt

BombDrop

The first three arguments are the origin of the dropped bomb. The last argument is set to 1 if the bomb has been planted. It is 0 if the bomb was dropped due to voluntary dropping or death/disconnect. Setting the last argument 1, will also trigger the round timer to hide. It also will show the dropped bomb on the Terrorist team's radar in the location specified by the first three arguments.

Name: Structure:	BombDrop	
	coord coord coord byte	CoordX CoordY CoordZ Flag

BombPickup

This message just tells the game that the bomb has been picked up. It will cause the dropped/planted bomb to disappear from the Terrorist team's radar.

Note: This message has no arguments.

Name:	BombPickup
Structure:	.

BotProgress

Used by CZ's bots when learning a new map. Displays a progress bar in the middle of the screen, with some header text. The bar doesn't move, and you can't do anything while the bar is displayed. This is a different style of progress bar than from the BarTime event. This really doesn't display well in CS.

Note: Flag can be 0 (update bar), 1 (create new bar), or 2 (remove bar). When using flag 0, send all arguments. When using flag 1, send only Flag and Header. When using flag 2, send only Flag.

Name:	BotProgress						
Structure:	<table> <tr> <td>byte</td><td>Flag</td></tr> <tr> <td>byte</td><td>Progress</td></tr> <tr> <td>string</td><td>Header</td></tr> </table>	byte	Flag	byte	Progress	string	Header
byte	Flag						
byte	Progress						
string	Header						

BotVoice

Displays (or hides) the voice icon above a user's head and the talking icon on the right side of the screen. This is called by CZ for bots; it's not called by regular players, although you *can* specify a regular player (non-bot) for the PlayerIndex. Status is 1 for talking, or 0 for not talking.

Name:	BotVoice				
Structure:	<table> <tr> <td>byte</td><td>Status</td></tr> <tr> <td>byte</td><td>PlayerIndex</td></tr> </table>	byte	Status	byte	PlayerIndex
byte	Status				
byte	PlayerIndex				

Brass

Creates a brass shell. Used, for example, by the AWP, after firing.

Name:	Brass
--------------	-------

Structure:	byte	Unknown
	coord	StartX
	coord	StartY
	coord	StartZ
	coord	VelocityX?
	coord	VelocityY?
	coord	VelocityZ?
	coord	UnknownX
	coord	UnknownY
	coord	UnknownZ
	angle	Life?
	short	Model?
	byte	Unknown
	byte	Unknown
	byte	Unknown

BuyClose

Forces the buy menu to close. This is not called when the player closes the buy menu by himself; it's only called when the game forces him to do so (ie: he walks outside of the buy zone, gets killed, etcetera).

Note: This message has no arguments.

Name:	BuyClose
Structure:	.

ClCorpse

Spawns a player's corpse. ModelName is the player's model name, for example: "leet". Delay is a delay before animation playback, can be negative value.

Note: Coord and Delay is in non-standard unknown format.

Note: In CS argument #10 is always equal to 0.

- See CS Team Constants (http://www.amxmodx.org/funcwiki.php?go=module&id=4#const_teams) for team indeces constants list.

Name:	ClCorpse	
Structure:	string	ModelName
	long	CoordX
	long	CoordY

long	CoordZ
coord	Angle0
coord	Angle1
coord	Angle2
long	Delay
byte	Sequence
byte	ClassID?
byte	TeamID
byte	PlayerID

Crosshair

Draws/removes a crosshair. If Flag is set to 1 the crosshair will be drawn.

Note: This crosshair looks not like the regular one but like the one that is drawn in the spectator mode.

Name:	Crosshair
Structure:	byte Flag

CurWeapon

Updates the clip ammo number and weapon's corresponding ammo type icon on the HUD.

Note: See CS Weapons Information for more information.

Name:	CurWeapon
Structure:	byte IsActive byte WeaponID byte ClipAmmo

CZCareer

Supplies certain updates to the player regarding Condition Zero singleplayer missions.

Note: See the CZCareer page for more information.

Note: Using an invalid type has no repercussions whatsoever. Therefore, you can use

this to make your own custom communications (you can send any number of arguments of any type after the first string).

Note: The Type argument is case-sensitive.

Note: This event does nothing in CS and CZ multiplayer.

Name: Structure:	CZCareer	
	string *	Type Parameters

CZCareerHUD

Displays certain HUD elements regarding Condition Zero singleplayer missions.

Note: See the CZCareerHUD page for more information.

Note: Using an invalid type has no repercussions whatsoever. Therefore, you can use this to make your own custom communications (you can send any number of arguments of any type after the first string).

Note: The Type argument is case-sensitive.

Note: This event has some limited functionality in CS and CZ multiplayer (albeit better in CZ).

Name: Structure:	CZCareerHUD	
	string *	Type Parameters

Damage

Called when a player takes damage, to display the red locational indicators. The last three arguments is the origin of the damage inflictor or victim origin if inflictor isn't found. DamageType is a bitwise value usually consist of a single bit.

Name: Structure:	Damage	
	byte	DamageSave
	byte	DamageTake
	long	DamageType
	coord	CoordX
	coord	CoordY

coord CoordZ

DeathMsg

Fired to all players (MSG_ALL or MSG_BROADCAST) to notify them of a death. This generates the HUD message the client sees in the upper right corner of their screen.

Also prints the console text message "KillerName killed VictimName with TruncatedWeaponName" or "*** KillerName killed VictimName with a headshot from TruncatedWeaponName ***"

Note: TruncatedWeaponName doesn't contain a "weapon_" prefix. See CS Weapons Information for more information.

Note: For grenade kill TruncatedWeaponName isn't "hegrenade" but "grenade", in fact it's an actual classname of a thrown grenade.

Note: For KillerID it's not necessarily should be a PlayerID, it could be 0 if player dies from fall/acid/radiation/fire/etc damage/lack of oxygen or from touch to a "trigger_hurt" entity, in this case TruncatedWeaponName will be "worldspawn" and "trigger_hurt" respectively.

Note: For vehicle kills TruncatedWeaponName could be "vehicle", "tank" et cetera.

Name: Structure:	DeathMsg	
	byte byte byte string	KillerID VictimID IsHeadshot TruncatedWeaponName

FlashBat

Updates the flashlight battery charge on the HUD.

Name: Structure:	FlashBat	
	byte	ChargePercents

Flashlight

Updates the flashlight state and battery charge on the HUD. If Flag is set to 1 the flashlight HUD icon will be

shown as active.

Name: Structure:	Flashlight	
	byte byte	Flag ChargePercents

Fog

Note: No Information available for this message.

ForceCam

Called whenever mp_forcecam or mp_forcechasecam are changed, with their new values passed. There is assumedly a third CVar that this tracks, but it is yet unknown. Note that this message doesn't actually change any of the spectating rules for the client.

Note: Even if mp_forcechasecam is set to 2, it is sent by this message as 1.

Name: Structure:	ForceCam	
	byte byte byte	ForcecamValue ForcechasecamValue Unknown

GameMode

Note: No Information available for this message.

Name: Structure:	GameMode	
	byte	Unknown

GameTitle

Note: No Information available for this message.

Geiger

Notifies about radiation level through special sound signals. Distance is a distance to hazard area.

Name:	Geiger
Structure:	byte Distance

Health

Updates the number of health on the HUD.

Name:	Health
Structure:	byte Health

HideWeapon

Hides the specified HUD elements.

Flags:

```

1  (1<<0) - crosshair, ammo, weapons list
2  (1<<1) - flashlight, +
4  (1<<2) - ALL
8  (1<<3) - radar, health, armor, +
16 (1<<4) - timer, +
32 (1<<5) - money, +
64 (1<<6) - crosshair
128 (1<<7) - +

```

Symbol + mean that an additional crosshair will be drawn. That crosshair looks exactly like the one from Crosshair message.

Name:	HideWeapon
Structure:	byte Flags

HLTV

Note: No Information available for this message.

Note: On new round is fired with both arguments equal to 0.

Name:	HLTV
--------------	------

Structure:	<table> <tr> <td>byte</td><td>ClientID</td></tr> <tr> <td>byte</td><td>Unknown</td></tr> </table>	byte	ClientID	byte	Unknown
byte	ClientID				
byte	Unknown				

HostageK

Temporarily draws a blinking red dot on the CT players' radar when a hostage is killed.

Name:	HostageK		
Structure:	<table> <tr> <td>byte</td><td>HostageID</td></tr> </table>	byte	HostageID
byte	HostageID		

HostagePos

Draws/updates the blue mark on the CT players' radar which is indicate the corresponding hostage position.

Note: Is called with Flag set to 1 on player HUD full update.

Name:	HostagePos										
Structure:	<table> <tr> <td>byte</td><td>Flag</td></tr> <tr> <td>byte</td><td>HostageID</td></tr> <tr> <td>coord</td><td>CoordX</td></tr> <tr> <td>coord</td><td>CoordY</td></tr> <tr> <td>coord</td><td>CoordZ</td></tr> </table>	byte	Flag	byte	HostageID	coord	CoordX	coord	CoordY	coord	CoordZ
byte	Flag										
byte	HostageID										
coord	CoordX										
coord	CoordY										
coord	CoordZ										

HudText

Note: No Information available for this message.

HudTextArgs

Prints HUD text.

Note: An example of TextCode could be "#Hint_you_have_the_bomb".

Note: If you have a problems with specifying the last two arguments use 1 and 0 respectively.

Name:	HudTextArgs						
Structure:	<table> <tr> <td>string</td><td>TextCode</td></tr> <tr> <td>byte</td><td>Unknown</td></tr> <tr> <td>byte</td><td>Unknown</td></tr> </table>	string	TextCode	byte	Unknown	byte	Unknown
string	TextCode						
byte	Unknown						
byte	Unknown						

HudTextPro

Note: No Information available for this message.

InitHUD

Initializes the HUD.

Note: This message has no arguments.

Name:	InitHUD
Structure:	.

ItemPickup

Temporarily draws correspodng item HUD icon in the middle of the right side of the screen.

Note: Draw time is depend on the *hud_drawhistory_time* client CVar value.

Name:	ItemPickup		
Structure:	<table> <tr> <td>string</td><td>ItemName</td></tr> </table>	string	ItemName
string	ItemName		

ItemStatus

Notifies about carried items.

Example of some item bits:

```

1  (1<<0) - nightvision goggles
2  (1<<1) - defusal kit

```

Name:	ItemStatus
--------------	------------

Structure:

byte	ItemsBitSum
-------------	-------------

Location

Note: No Information available for this message.

Name:

Location

Structure:

byte	Unknown
string	Unknown

Money

Updates the Amount of money on the HUD. If the Flag is 1, the additional new-old HUD money amount difference will be also displayed.

Name:

Money

Structure:

long	Amount
byte	Flag

MOTD

Displays MOTD window.

Note: Max. Text length is 60. Large MOTD is sent in multiple messages. For the message that is send a last MOTD part the Flag will be set to 1 and 0 otherwise.

Name:

MOTD

Structure:

byte	Flag
string	Text

NVGToggle

Toggles night vision mode. For Flag: 1 is on, 0 is off.

Name:	NVGToggle		
Structure:	<table> <tr> <td>byte</td><td>Flag</td></tr> </table>	byte	Flag
byte	Flag		

Radar

Draws/updates the dot on the HUD radar which is indicate the given player position.

Note: Works for teammates only.

Name:	Radar								
Structure:	<table> <tr> <td>byte</td><td>PlayerID</td></tr> <tr> <td>coord</td><td>CoordX</td></tr> <tr> <td>coord</td><td>CoordY</td></tr> <tr> <td>coord</td><td>CoordZ</td></tr> </table>	byte	PlayerID	coord	CoordX	coord	CoordY	coord	CoordZ
byte	PlayerID								
coord	CoordX								
coord	CoordY								
coord	CoordZ								

ReceiveW

Note: No Information available for this message.

ReloadSound

Note: No Information available for this message.

Name:	ReloadSound				
Structure:	<table> <tr> <td>byte</td><td>Unknown</td></tr> <tr> <td>byte</td><td>Unknown</td></tr> </table>	byte	Unknown	byte	Unknown
byte	Unknown				
byte	Unknown				

ReqState

Note: No Information available for this message.

Note: This message has no arguments.

Name:	ReqState
--------------	----------

Structure:

ResetHUD

Resets the HUD.

Note: This message has no arguments.

Name:	ResetHUD
Structure:	.

RoundTime

Updates the round timer on the HUD. Time is in seconds.

Name:	RoundTime		
Structure:	<table><tr><td>short</td><td>Time</td></tr></table>	short	Time
short	Time		

SayText

Prints say HUD text. Second argument can be a predefined string or a custom one. In the last case the last two arguments isn't required.

An examples of a predefined Counter-Strike string values: #Cstrike_Chat_AllDead, #Cstrike_Name_Change

Note: For #Cstrike_Name_Change String2 is an old name and String3 is a new name.

Name:	SayText								
Structure:	<table><tr><td>byte</td><td>SenderID</td></tr><tr><td>string</td><td>String1</td></tr><tr><td>string</td><td>String2</td></tr><tr><td>string</td><td>String3</td></tr></table>	byte	SenderID	string	String1	string	String2	string	String3
byte	SenderID								
string	String1								
string	String2								
string	String3								

Scenario

If Active is 0, this display will be hidden. If Active is 1, displays Sprite (valid names listed in sprites/hud.txt)

to the right of the round timer with an alpha value of Alpha (100-255). If FlashRate is nonzero, then the sprite will flash from the given alpha to an alpha of 100, at a rate of FlashRate (measured in ???). This is used by CZ to display how many hostages remain unrescued, and also to display the ticking bomb when it is planted.

Note: If Active is 0, don't send any other arguments afterwards. Also, you don't need to send either short if FlashRate is just going to be 0.

Note: Works in both CS and CZ!

Note: In CZ (and possibly CS), if someone respawns after the bomb has been planted, their Scenario event will not work at all until the next round.



Name:	Scenario										
Structure:	<table> <tr> <td>byte</td><td>Active</td></tr> <tr> <td>string</td><td>Sprite</td></tr> <tr> <td>byte</td><td>Alpha</td></tr> <tr> <td>short</td><td>FlashRate</td></tr> <tr> <td>short</td><td>Unknown</td></tr> </table>	byte	Active	string	Sprite	byte	Alpha	short	FlashRate	short	Unknown
byte	Active										
string	Sprite										
byte	Alpha										
short	FlashRate										
short	Unknown										

ScoreAttrib

Updates the scoreboard's attribute for the specified player. For the 2nd argument, 0 is nothing, (1<<0) i.e. 1 is dead, (1<<1) i.e. 2 is bomb, (1<<2) i.e. 4 is VIP.

Note: Flags is a bitwise value so if VIP player is dying with the bomb the Flags will be 7 i.e. bit sum of all flags.

Name:	ScoreAttrib				
Structure:	<table> <tr> <td>byte</td><td>PlayerID</td></tr> <tr> <td>byte</td><td>Flags</td></tr> </table>	byte	PlayerID	byte	Flags
byte	PlayerID				
byte	Flags				

ScoreInfo

Updates the scoreboard with the given player's Frags and Deaths.

Note: In CS the 4th argument is always equal to 0.

- See CS Team Constants (http://www.amxmodx.org/funcwiki.php?go=module&id=4#const_teams) for team indices constants list.

Name:	ScoreInfo
--------------	-----------

Structure:	byte	PlayerID
	short	Frag
	short	Deaths
	short	ClassID?
	short	TeamID

ScreenFade

Fades the screen.

Note: Duration and HoldTime is in special units. 1 second is equal to (1<<12) i.e. 4096 units.

Flags (from HLSDK):

```
FFADE_IN          0x0000 // Just here so we don't pass 0 into the function
FFADE_OUT         0x0001 // Fade out (not in)
FFADE_MODULATE    0x0002 // Modulate (don't blend)
FFADE_STAYOUT     0x0004 // ignores the duration, stays faded out until new ScreenFade message received
```

Name:	ScreenFade	
Structure:	short	Duration
	short	HoldTime
	short	Flags
	byte	ColorR
	byte	ColorG
	byte	ColorB
	byte	Alpha

ScreenShake

Shakes the screen.

Note: All arguments is in special units. 1 second is equal to (1<<12) i.e. 4096 units.

Name:	ScreenShake	
Structure:	short	Amplitude
	short	Duration
	short	Frequency

SendAudio

Plays the specified audio. An example of AudioCode could be "%!MRAD_rounddraw".

Name:	SendAudio						
Structure:	<table><tr><td>byte</td><td>SenderID</td></tr><tr><td>string</td><td>AudioCode</td></tr><tr><td>short</td><td>Pitch</td></tr></table>	byte	SenderID	string	AudioCode	short	Pitch
byte	SenderID						
string	AudioCode						
short	Pitch						

ServerName

Sends a server name to a client.

Name:	ServerName		
Structure:	<table><tr><td>string</td><td>ServerName</td></tr></table>	string	ServerName
string	ServerName		

SetFOV

Sets the specified field of view.

Name:	SetFOV		
Structure:	<table><tr><td>byte</td><td>Degrees</td></tr></table>	byte	Degrees
byte	Degrees		

ShadowIdx

Note: No Information available for this message.

Name:	ShadowIdx		
Structure:	<table><tr><td>long</td><td>Unknown</td></tr></table>	long	Unknown
long	Unknown		

ShowMenu

Displays a "menu" to a player (text on the left side of the screen). Acts like AMXX's show_menu (in fact, this is how AMXX shows a menu).

Note: Multipart should be 1 if your menu takes up multiple messages (i.e.: string is too big to fit into one). On the final message, Multipart should be 0.

Name: Structure:	ShowMenu	
	short char byte string	KeysBitSum Time Multipart Text

ShowTimer

Forces the round timer displaying.

Note: This message has no arguments.

Name: Structure:	ShowTimer .
-----------------------------------	----------------

SpecHealth

Note: No Information available for this message.

Name: Structure:	SpecHealth	
	byte	Health?

SpecHealth2

Updates the observer's screen with a name and health of the given player.

Name: Structure:	SpecHealth2	
	byte byte	Health PlayerID

Spectator

Called when player becomes an observer/spectator.

Note: On join to Spectators usually is fired twice in a row.

Name:	Spectator				
Structure:	<table> <tr> <td>byte</td><td>ClientID</td></tr> <tr> <td>byte</td><td>Unknown</td></tr> </table>	byte	ClientID	byte	Unknown
byte	ClientID				
byte	Unknown				

StatusIcon

Draws/removes the specified status HUD icon. For Status, 0 is Hide Icon, 1 is Show Icon, 2 is Flash Icon. Color arguments are optional and is required only if Status aren't equal to 0.

Name:	StatusIcon										
Structure:	<table> <tr> <td>byte</td><td>Status</td></tr> <tr> <td>string</td><td>SpriteName</td></tr> <tr> <td>byte</td><td>ColorR</td></tr> <tr> <td>byte</td><td>ColorG</td></tr> <tr> <td>byte</td><td>ColorB</td></tr> </table>	byte	Status	string	SpriteName	byte	ColorR	byte	ColorG	byte	ColorB
byte	Status										
string	SpriteName										
byte	ColorR										
byte	ColorG										
byte	ColorB										

StatusValue

Sends/updates the status values. For the Flag 1 the Value is TeamRelation, for 2 is PlayerID, for 3 is Health. For TeamRelation, 1 is Teammate player, 2 is Non-Teammate player, 3 is Hostage. If TeamRelation is Hostage, PlayerID will be 0 or will be not sent at all.

Usually is fired as a triple message, for example:

```
{1, 2} - non-teammate player
{2, 17} - player index is 17
{3, 59} - player health is 59
```

Name:	StatusValue				
Structure:	<table> <tr> <td>byte</td><td>Flag</td></tr> <tr> <td>short</td><td>Value</td></tr> </table>	byte	Flag	short	Value
byte	Flag				
short	Value				

StatusText

Specifies the status text format.

Name: Structure:	StatusText	
	byte string	Unknown Text

TaskTime

Displays a secondary timer above the round timer. Used for Condition Zero singleplayer missions.

If Time is -1, timer disappears. If Time is any other negative value, it is displayed as green instead of yellow, and considered positive.

If Active is true, timer counts down. Otherwise, it is paused.

If Fade is above zero, the timer will slowly fade out after that many seconds have passed (even if the timer is inactive).

Note: This event can only be used on missions that have an objective requiring a secondary timer!

Name: Structure:	TaskTime	
	short byte byte	Time Active Fade

TeamInfo

Sets the team information for the given player.

Note: In CS TeamName is either "UNASSIGNED", "TERRORIST", "CT" or "SPECTATOR".

Name: Structure:	TeamInfo	
	byte string	PlayerID TeamName

TeamScore

Updates the team score on the score board.

Note: In CS TeamName is either "TERRORIST" or "CT".

Name:	TeamScore		
Structure:	<table> <tr> <td>string short</td><td>TeamName Score</td></tr> </table>	string short	TeamName Score
string short	TeamName Score		

TextMsg

Prints a custom/or predefined text message.

There does not necessarily have to be a total of 6 arguments, there could be as little as 2. For example you can send a message with the following:

```
Arg1: 1
Arg2: #Game_join_ct
Arg3: Pimp Daddy
```

Name:	TextMsg												
Structure:	<table> <tr> <td>byte</td><td>DestinationType</td></tr> <tr> <td>string</td><td>Message</td></tr> <tr> <td>string</td><td>Submsg</td></tr> <tr> <td>string</td><td>Submsg</td></tr> <tr> <td>string</td><td>Submsg</td></tr> <tr> <td>string</td><td>Submsg</td></tr> </table>	byte	DestinationType	string	Message	string	Submsg	string	Submsg	string	Submsg	string	Submsg
byte	DestinationType												
string	Message												
string	Submsg												
string	Submsg												
string	Submsg												
string	Submsg												

Train

Displays the speed bar used for controlling a train.

Note: Speed is as follows: 0 (disable display), 1 (reverse), 2 (neutral), 3 (slow speed), 4 (medium speed), 5 (maximum speed)

Name:	Train		
Structure:	<table> <tr> <td>byte</td><td>Speed</td></tr> </table>	byte	Speed
byte	Speed		

TutorClose

Closes all CZ-style tutor popups.

TutorLine

Note: No Information available for this message.

TutorState

Note: No Information available for this message.

TutorText

Used to display a CZ-style tutor popup.

Name: Structure:	TutorText	
	string	Unknown
	byte	Unknown
	short	Unknown
	short	Unknown
	short	Unknown

ViewMode

Note: No Information available for this message (HLSDK says this switches to first-person view, but it doesn't seem to function as so).

Note: This message has no arguments.

Name: Structure:	ViewMode
	.

VGUIMenu

Displays a predefined VGUI menu.

Name: Structure:	VGUIMenu	
	byte	MenuID
	short	KeysBitSum

	<div>char byte string</div> <div>Time? Multipart? Name?</div>
--	---

VoiceMask

Used to tell a client who he can hear over the microphone.

Name:	VoiceMask
Structure:	<div>long long</div> <div>AudiblePlayersIndexBitSum ServerBannedPlayersIndexBitSum</div>

WeaponList

Configures the HUD weapons list.

Note: Fired on map initialization.

Note: SlotID starts from 0.

Flags (from HLSDK):

```

ITEM_FLAG_SELECTONEMPTY      1
ITEM_FLAG_NOAUTORELOAD      2
ITEM_FLAG_NOAUTOSWITCHEMPTY  4
ITEM_FLAG_LIMITINWORLD      8
ITEM_FLAG_EXHAUSTIBLE      16 // A player can totally exhaust their ammo supply and lose this weapon.

```

Note: See CS Weapons Information for more information.

Name:	WeaponList
Structure:	<div>string byte byte byte byte byte byte byte</div> <div>WeaponName PrimaryAmmoID PrimaryAmmoMaxAmount SecondaryAmmoID SecondaryAmmoMaxAmount SlotID NumberInSlot WeaponID Flags</div>

WeapPickup

Temporarily draws corresponding weapon HUD icon in the middle of the right side of the screen.

Note: Draw time is depend on the `hud_drawhistory_time` client CVar value.

Note: Is fired right before weapon is picked up (notice `"before"`).

Note: See CS Weapons Information for more information.

Name:	WeapPickup		
Structure:	<table><tr><td>byte</td><td>WeaponID</td></tr></table>	byte	WeaponID
byte	WeaponID		

Retrieved from "http://wiki.alliedmods.net/Half-Life_1_Game_Events"

Categories: Half-Life 1 | Scripting (AMX Mod X)

-
- This page was last modified 22:28, 18 February 2007.