Half-Life Asset Manager Manual

Version 2.1.0

Sam Vanheer

January 15, 2024

Table Of Contents

1 About Half-Life Asset Manager

Half-Life Asset Manager is a tool for the Half-Life 1 engine, also known as the GoldSource engine. It is also compatible to some extent with some offshoots like Xash and Svengine.

Asset Manager supports the loading, viewing and editing of Half-Life 1 Studio models.

It does not support Source 1 and Source 2, though some file formats are recognized for the purpose of redirecting users to other programs.

1.1 Known Issues

There are some bugs that will be fixed in a future major version due to depending on the implementation of a new shader-based renderer.

These issues are:

- **Mirror Model** does not work properly when the ground has been moved up or down (mirroring is done using a faked method, modern methods can do this properly)
- Shadows are always drawn at a height of 1 unit above the world origin instead of 1 unit above the ground

2 Installing

Asset Manager is available in installer and archive form. The installer will automatically install the program to a location of your choosing and can optionally create shortcuts for you.

When using the off-line installer you must uninstall the previous version before installing a new version.

The archive version does not require installation itself but does require you to install the Visual C++ redistributable included with the archive. This redistributable includes the Visual C++ runtime used by the program and is required to run it.

3 General Usage

When installing Asset Manager using the installer you are given the option to associate the program with file types supported by the tool. Enabling this allows you to open Asset Manager by double clicking the file in Windows Explorer.

You can also associate file types manually by following this tutorial: https://www.thewindowsclub.com/change-file-associations-windows

Asset Manager treats positional arguments passed on the command line as a filename. Only one filename may be passed and it should be enclosed by quotes. If passed without quotes Asset Manager will attempt to reconstruct the filename by joining all positional arguments together separated by a space.

Parts of the user interface allow editing numeric data using spin boxes like these:
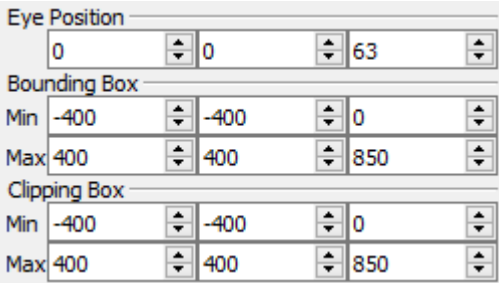
Clicking the up and down arrow buttons adjusts these values by one. Clicking them while holding down the **CTRL** key is the equivalent of clicking them **10** times. This can be used to quickly make large adjustments.

4 Portable Mode

Launching Half-Life Asset Manager with the command line parameter `--portable` launches it in portable mode. In this mode the program configuration file is stored next to the executable, which makes it easier to run the program from a removable storage device or with multiple user accounts.

5 Configuring Log Output

Half-Life Asset Manager outputs log information to a file.

Log output is written to a text file called `HLAM-Log.txt` located in the same directory as the program configuration file.

When running the program using its default configuration this will be in `%AppData%/Roaming/Half-Life Asset Manager` (Windows). When running in portable mode this will be in the directory containing `HLAM.exe`.

You can open this directory by opening the options dialog and clicking `Open Configuration Directory`.

The file is cleared on startup.

5.1 Configuring Logging Categories

Qt uses logging categories to filter log output. To control which categories are enabled, create a file called `qtlogging.ini` in `Half-Life Asset Manager` (next to the `HLAM` executable) and add the following:

```
[Rules]
```

Now add logging rules below this section, for example:

```
[Rules]
qt.qpa.gl=true
```

This will enable the logging category for OpenGL.

6 Supported Asset Types

6.1 Half-Life 1 StudioModel

The Half-Life 1 StudioModel format identified as IDST version 10 is fully supported.

See Section Half-Life 1 StudioModel for more information.

6.2 Source 1 StudioModel

The Source 1 StudioModel format identified as IDST versions 44 through 48 are recognized and forwarded to a user-configured Source 1 Model Viewer.

See Section Supported External Programs for more information.

6.3 Xash StudioModel

The Xash StudioModel format identified as IDST version 10 with additional Xash-specific data is recognized and forwarded to a user-configured Xash Model Viewer if configured in the Studiomodel options page.

If a Xash model is loaded in HLAM itself and then saved any Xash-specific data included in the model will be lost. HLAM warns users about this before saving.

See Section Supported External Programs for more information.

6.4 Counter-Strike Nexon StudioModel

The Counter-Strike StudioModel format identified as IDST version 11 with additional Nexon-specific data is recognized and forwarded to a user-configured Counter-Strike Nexon Model Viewer.

See Section Supported External Programs for more information.

6.5 Quake 1 Alias Model

The Quake 1 Alias Model format identified as IDPO is recognized and forwarded to a user-configured Quake 1 Model Viewer.

See Section Supported External Programs for more information.

7 Menus

7.1 File

7.1.1 Load...

Load one or more assets from files. This operation can take a long time if there are many assets.

Each asset is capable of resolving file paths based on where the asset file was loaded from. See section Game Configurations for more information.

7.1.2 Save

Saves the current asset to the file it was loaded from.

If the asset was loaded from a file format that is only supported for importing the asset will be renamed to the preferred extension.

If the new name is in use by another loaded asset the user will be asked whether to discard the other asset or cancel the save operation.

7.1.3 Save As...

Saves the current asset to a file of the user's choosing.

If the new name is in use by another loaded asset the user will be asked whether to discard the other asset or cancel the save operation.

7.1.4 Close

Closes the current asset.

The user will be asked to save or discard any unsaved changes made to the assets. The user can also cancel closing.

7.1.5 Close All

Closes all loaded assets.

The user will be asked to save or discard any unsaved changes made to the assets.

7.1.6 Recent Files

Shows the list of recently opened files.

7.1.7 Exit

Closes the program. The user will be asked to save or discard any unsaved changes made to opened assets.

7.2 Edit

7.2.1 Undo

Undo the last change made to an asset.

7.2.2 Redo

Redo the last undone change made to an asset.

7.3 View

7.3.1 Fullscreen

When one or more assets are loaded this action allows the asset to be viewed in fullscreen mode.

If the asset shows a 3D scene only the scene will be displayed in the fullscreen window.

Fullscreen mode can be toggled using **F11**.

7.3.2 File Browser

The File Browser displays the contents of a directory and allows files to be opened directly. The browser has a filter option to show only files that match a specific file format and filters out files that cannot be opened (e.g. StudioModel texture files that belong to another StudioModel file).

Using the arrow and Enter keys allows one to quickly view models.

Note that using a format filter can be slow if there are many files in a directory since each file has to be opened to detect its format.

The root directory can be changed at will and can be set to the directory belonging to a game configuration to quickly switch directories.

Double-clicking directories switches the root directory to that directory. Double-clicking the parent directory `..` switches to the parent directory.

Multiple files can selected by clicking and holding the left mouse button and dragging over the filenames, using `CTRL+Left Mouse Button` or by using `CTRL+A` to select all files.

Multiple files can be opened by selecting the files and clicking `Open Selected`.

7.3.3 Messages

The Messages panel shows debug, info and error messages emitted by the program.

By default only info and error messages are shown. The message category can be enabled to get more information about where a message came from.

This panel is automatically opened when an error occurs.

7.4 Audio

7.4.1 Play Sounds

Whether sounds are played when needed.

7.4.2 Framerate Affects Pitch

When enabled the framerate that an asset is displayed at affects the sound pitch.

7.5 Video

7.5.1 Power Of 2 Textures

Whether textures are rescaled to a power of 2. Older graphics cards don't support non-power of 2 textures.

7.5.2 Wait For Vertical Sync

Also known as VSync.

Whether to enable vertical sync on the 3D window. Enabling vertical sync clamps the framerate to the display's refresh rate.

Disabling it allows for higher framerates but may cause screen tearing.

7.5.3 Min Filter

The minification filter used for textures.

See https://www.khronos.org/opengl/wiki/Sampler_Object#Filtering

7.5.4 Mag Filter

The magnification filter used for textures.

See https://www.khronos.org/opengl/wiki/Sampler_Object#Filtering

7.5.5 Mipmap Filter

The mipmap filter used for textures.

See https://www.khronos.org/opengl/wiki/Sampler_Object#Filtering

7.5.6 Anti-Aliasing

The Multisample Anti-aliasing setting. Setting this to a higher value reduces jagged edges but requires a more powerful graphics card.

Changing this setting recreates the 3D window.

7.5.7 Transparent Screenshots

Whether screenshots are transparent or include the background color.

See Section Take Screenshot for more information.

7.5.8 Take Screenshot…

Takes a screenshot and asks the user where to save it. The keyboard shortcut **F10** can also be used to take a screenshot.

This action is affected by the setting of the Video->Transparent Screenshots action.

Transparent screenshots exclude the background from the resulting image. Note however that transparent objects will still be affected by the background color.

To avoid this set the background color to black first using Tools->Options->Colors->Background.

This will cause transparent objects to be rendered using only their own color and not that of the background.

7.6 Asset

This section in the menu bar is reserved for the asset-specific Asset menu. Actions in this menu affect the current asset, so its contents depend on the type of the asset.

Consult the documentation for Section Supported Asset Types for more information.

7.7 Tools

7.7.1 Refresh

Reloads the current asset. The user will be asked to save or discard any changes made to the asset, or cancel the operation.

7.7.2 Asset-specific menus

This section in the Tool menu is reserved for asset-specific menus.

Consult the documentation for Section Supported Asset Types for more information.

7.7.3 Options

Opens the Options dialog.

7.8 Help

7.8.1 Open Manual…

Opens this manual in the system-configured pdf viewer.

7.8.2 About

Shows the About dialog containing information about the program version, author, build date and more.

7.8.3 About Qt

Shows the About Qt dialog containing information about the Qt version used as well as license information.

8 Options Dialog

The Options dialog is divided into categories. Categories have one or more pages.

Asset types can provide additional options categories and pages. Consult the documentation for Section Supported Asset Types for more information.

8.1 General

8.1.1 Pause animations when clicking on the timeline

Clicking on the timeline pauses animations to make seeking a specific frame easier.

8.1.2 Allow tab close with middle click

Allow tabs to be closed by middle clicking them. This behaves as though the tab's close button was clicked.

8.1.3 Limit to one open asset at a time

When enabled opening an asset will close any currently opened asset. This is convenient for quickly checking files without keeping them opening longer than necessary.

8.1.4 Maximum Recent Files

The maximum number of recent files to show in the Recent Files menu.

8.1.5 Tick Rate

Also known as the framerate or frames per second (FPS).

The amount of times per second to update the current asset. Higher values allow for smoother playback but has higher CPU usage.

8.1.6 Invert X Axis

Invert the X axis when interacting with the 3D window.

8.1.7 Invert Y Axis

Invert the Y axis when interacting with the 3D window.

8.1.8 Mouse Sensitivity

Controls the mouse sensitivity. The higher the sensitivity the more pronounced changes become.

8.1.9 Mouse Wheel Speed

How fast the mouse wheel affects the scene.

8.1.10 Enable Audio Playback

Whether to enable audio playback. When disabled no attempt is made to access the computer's audio devices.

8.1.11 Mute Application When Application Is In Background

If enabled all audio is muted when the application is not active (e.g. minimized, behind another program, etc).

8.2 Style

Allows the program style sheet to be replaced with a user-defined alternative. This allows the programs to be set to dark mode, among other things.

8.3 Colors

Allows colors used in various parts of the program to be changed.

8.4 External Programs

External programs are launched by Asset Manager when needed to complete certain tasks such as compiling models or opening models in third party model viewers.

8.4.1 Prompt before launching external programs

For actions that automatically choose to use external programs this option requires those actions to ask the user to continue.

8.4.2 Supported External Programs

Note: model viewers need to support the loading of models by passing the filename on the command line to work as an external program.

Users can specify additional command line arguments if needed to correctly launch programs. It is also possible to use batch files by launching `cmd.exe` with the `/C` argument to allow a script to handle the launching of an external program.

The working directory is set to the location of the program executable. When using scripts it may be necessary to change this to the location of the script to ensure correct execution.

- Counter-Strike Nexon Model Viewer: Path to a model viewer that can display Counter-Strike Nexon Studio models

  No known model viewers support this format

- Quake 1 Model Viewer: Path to a model viewer that can display Quake 1 Alias models

  Known model viewers:

  - [Quake 1 Model Viewer](#)

- Source 1 Model Viewer: Path to a model viewer that can display Source 1 models

  Known model viewers:

  - Source 1 `hlmv.exe` included with Source 1 games. Prefer using a model viewer running on the latest engine branch to ensure all 4 versions of the format can be opened. You may need to add this additional command line argument to point the program to the correct game directory: `-game "path/to/gameinfo.txt/directory"`.

- Crowbar: Path to the Crowbar tool

  More information and a download link can be found here: https://steamcommunity.com/groups/CrowbarTool

- Xash Model Viewer: Path to a model viewer that can display Xash Studio models

  Known model viewers:

  - [Paranoia 2 Model Viewer](#)

8.5 Assets

This category is populated by asset type-specific pages.

Consult the documentation for Section Supported Asset Types for more information.

8.6 Game Configurations

Add game configurations to enable Asset Manager to locate related game assets. The configuration is auto-detected based on filename.

If no configuration could be found Asset Manager will check if the asset is in a game asset directory (ex: models) and creates a temporary configuration based on that directory structure. The default configuration is always used as a fallback if no configuration could be found, even if a temporary one was created.

If a game's `liblist.gam` file contains a `fallback_dir` entry it will be used to find that game's configuration and use it as well as the originating game's configuration. This matches the engine's behavior.

Changing configurations will automatically reinitialize the filesystem for all loaded assets. The same happens if an asset is renamed. Only files loaded after the filesystem has reinitialized will be affected by this change.

8.6.1 File System

Allows filesystem behavior to be configured. Users can control whether game directories containing additional game content like HD or Addon content are searched.

8.6.1.1 Steam Language

The language that Steam is configured to use. This setting affects how files referenced by assets are located to allow localized files to be used instead.

8.6.2 Auto-Detect Game Configurations

The Auto-Detect Game Configurations wizard allows game configurations to be added automatically.

Set the path to the game directory, select the game executable and base game directory and the wizard will scan for mods. All mods that provide a `liblist.gam` file are detected and their name is extracted from the file.

You can then select the mods you want to add as configurations, move them to the list of configurations to add and complete the process.

9 Half-Life 1 StudioModel

9.1 Asset Load Behavior

Loading Half-Life 1 Studio models that have sequence group and/or external texture files will show information in the Messages panel indicating that these files have been merged into the main file. Saving the file through Asset Manager will save the merged file.

9.2 Asset Menu

9.2.1 Dock Widgets

Provides actions toggle each of the dock widgets in the user interface.

9.2.2 Reset Dock Widgets

Resets the dock widgets to their initial visibility state and position.

9.2.3 Show Controls Bar

Toggles the control bar located directly under the 3D window.

9.2.4 Show Timeline

Toggles the timeline located at the bottom of the edit controls.

9.2.5 Show Edit Controls

Toggles between the full edit controls view and a maximized 3D window view.

Hiding the edit controls also hides the controls bar and timeline, so those actions are disabled while hidden.

9.2.6 Previous Camera

Cycles to the previous camera type in the scene.

9.2.7 Next Camera

Cycles to the next camera type in the scene.

9.2.8 Center View

Centers the current camera on the selected axis. Not all cameras support centering.

9.2.9 Save View

Saves the current camera and camera state.

9.2.10 Restore View

Restores the saved camera and camera state.

This works even when the saved view is restored with another asset.

9.2.11 Flip Normals

Flips the vertex normals in all meshes.

9.2.12 Show QC Data

Opens the QC data dialog. This dialog shows the model data in the QC format.

Because not all of the original QC data is available this is only pseudo-QC data. Use a decompiler to obtain QC data that can be used to compile a model.

9.2.13 Dump Model Info…

Dumps model info to a file of the user's choosing.

9.3 Tool Menu

The StudioModel Tool menu adds actions for interacting with StudioModel files outside the program.

9.3.1 Launch Crowbar

Launches Crowbar. This action is only enabled if the path to the Crowbar tool has been specified in the External Programs options page.

See Section Supported External Programs for more information.

9.3.2 Edit QC File…

Opens the selected QC file in the system-configured text editor.

9.4 Options Dialog

The StudioModel page is added to provide options specific to this format.

9.4.1 Automatically switch to the First Person camera for view models

When enabled models whose name starts with v_ or V_ are loaded with the First Person camera activated.

9.4.2 Activate the Texture view when the Textures panel is opened

When enabled, when the Textures panel is made visible the Textures view is automatically selected.

This matches the behavior of earlier model viewers.

Note that the Textures view will also be activated if you drag the Textures panel to float it or dock it elsewhere.

9.4.3 Ground Length

The length of the ground in units.

9.4.4 Open Xash models in Xash model viewer

This setting determines how Xash models are loaded. **Ask** prompts users every time, **Always** and **Never** automatically use the chosen action.

9.4.5 Sound events

List of event Ids that should play sounds.

The following Ids are used in Half-Life: 1004, 1008, 5004.

The following Ids are used in Opposing Force: 1011, 1012, 1013.

They are automatically recognized.

9.5 User Interface

The StudioModel interface is collectively referred to as the Edit Controls.

It consists of the view, below that is the Controls Bar. Surrounding those are the dock widgets. Below those is the Timeline.

9.5.1 View

The view contains the Scene and Texture views.

The Scene view shows the 3D scene containing the model and other objects used for display purposes. You can left click and drag to rotate the model and right click and drag to zoom in and out.

Holding the Control key and left click dragging changes the light vector X and Y values. Holding the Control and Shift keys and left click dragging horizontally changes the light vector Z value.

Holding the Shift key or the Mouse Extra button 1 (usually the back button) while using the Arc Ball camera and left click dragging moves the camera's orbit point.

The Texture view shows the currently selected texture. You can left click and drag the texture and right click and drag to zoom in and out.

The Texture view is synchronized to the Textures panel.

9.5.2 Controls Bar

The Controls Bar contains the current frames per second, the drawn polygons count, the view selection tab bar and the pose selection dropdown.

The pose selection switches between Sequences (shows the currently selected animation) and Skeleton (shows the model as positioned without any animations affecting it).

9.5.3 Timeline

The Timeline shows the current animation frame and provides controls to restart, skip one frame back, skip 5 frames back, pause/play the animation, skip 5 frames ahead and skip one frame ahead.

It also provides the framerate control to speed up or slow down animation playback.

9.6 Dock Widgets

9.6.1 Cameras

The Cameras panel allows to change the current camera as well as modify the camera properties.

Three cameras are provided: **Arc Ball**, **Free Look** and **First Person**.

All cameras allow to change the field of view.

The **Arc Ball** camera orbits around a point in 3D space and allows zooming in and out.

The user can change the projection mode from perspective to orthographic. The camera attempts to match the orthographic zoom to roughly match the equivalent perspective distance. This compensation becomes less accurate if the 3D window aspect ratio is less like the display aspect ratio.

The **Free Look** camera allows free movement in 3D space. This allows the user to view a model from any angle.

The **First Person** camera shows models from the first person perspective and is useful mostly for weapon viewmodels (v_ models).

9.6.2 Scene

The Scene panel allows objects in the scene to be modified.

The Model object can be moved around. This does not affect the model data and is purely for display purposes.

The **Center On World Origin** button moves the model to position 0 0 0. The **Align On Ground** button moves the model up so that its sequence bounding box lies flat on the ground.

The Ground object allows the ground to be displayed. The model can be mirrored on the ground and a texture can be set.

By default the texture will be stretched to cover the ground. Checking the Enable Tiling checkbox instead causes the texture to be resized to the specified Tile Size.

The Background object allows a background texture to be displayed. This texture is stretched to fit the viewport.

9.6.3 Model Display

The Model Display panel allows various display settings to be changed.

The Render Mode allows to switch between **Wireframe** (only shows the polygon edges), **Flat Shaded** (lighting is not interpolated between vertices), **Smooth Shaded** and **Texture Shaded** modes.

The **Opacity** slider allows to change the model to render transparently.

Backface Culling hides the back face of polygons.

Wireframe Overlay adds a wireframe overlay on top of the model.

Draw Shadows shows the primitive shadow rendering technique that was once available in the engine. By default this will cause Z fighting due to how shadows are rendered.

Fix Shadow Z Fighting fixes the Z fighting issues.

Show Hitboxes shows the model's hitboxes. Setting the **HitboxFace** color in the Color options page and setting its alpha value to a value greater than 0 shows the hitbox faces.

Show Bones shows the bones and the connections between them.

Show Attachments shows the attachment points.

Show Eye Position shows the eye position, used by many NPCs to determine whether they can see other entities.

Show Bounding Box shows the bounding box used for the movement hull. Most models do not set this since it's only needed for large models that may be skipped if their origin is not visible.

Show Clipping Box shows the clipping box used for visibility culling. Most models do not set this since it's only needed for large models that may be skipped if their origin is not visible.

Show Normals shows the vertex normals.

Show Axes shows the world axes. X is red, Y is green and Z is blue.

Show Player Hitbox shows the in-game player hitbox size, aligned to the ground. Use the Scene->Model->Align On Ground button to align player models to this box.

Mirror on X/Y/Z Axis mirrors the model on the selected axes. Mirror On Y Axis is very useful for displaying Counter-Strike viewmodels as right-handed.

Show Crosshair shows a crosshair drawn in the center of the 3D window.

Show Guidelines shows a line on the drawn in the bottom center half of the 3D window. It also draws lines showing the edges of the visible space as it would be in-game.

Show Off-screen Areas shows the sections of the 3D window that are normally invisible in-game.

Aspect Ratio sets the aspect ratio used by **Show Guidelines** and **Show Off-screen Areas**. When you resize the window the program will adjust both to maintain the selected aspect ratio.

9.6.4 Lighting

The Lighting panel allows to control the lights in the scene.

The Sky Light object controls the environmental light. The pitch and yaw values are converted to a directional vector.

The user can change the light color on a per-asset basis here. The initial color is determined by the value of the SkyLight color set in the Color options page.

The ambient and shade lighting values are configurable. In-game these values are calculated by the engine based on the sky light, dynamic lights and either the floor or ceiling depending on whether the entity using the model has set the EF_INVLIGHT effects flag. Because this information isn't present here the values are configurable.

9.6.5 Sequences

The Sequences panel allows to preview and modify the model's sequences (animations).

Animations are referenced by the game in a couple different ways. NPCs commonly use activities to pick an animation but may also use the name or part of it. The Human Grunt for instance selects the stance used to aim and fire weapons using a prefix like `crouching_`.

Weapons in particular use the animation index, displayed in parentheses in the sequence list.

A special note for the Tentacle monster (`monster_tentacle`): this NPC uses custom activities. The program recognizes these and supports the use of custom activities just like the compiler does using the syntax `ACT_<number>` where `<number>` is an integer value. This value can be a negative value though the game only uses positive values.

The looping mode dropdown controls how the animation is looped. Always Loop causes it to loop no matter what, Never Loop only plays once, and Default uses the animation setting controlled by the Is Looping checkbox. This last option is what the game uses.

The blends controls can be used to preview an animation's blending behavior. Blending is typically used to allow the player or NPCs to turn their torso without turning the entire model.

Counter-Strike uses a 9-way blend that allows for more elaborate blending.

The animation framerate (FPS) controls how quickly the animation plays. Higher values speed up playback. A value of 0 causes it to freeze. Note that a value of 0 may cause the game to malfunction or crash. Use with caution.

The activity can be selected from the list of predefined activities, or a custom activity can be added using the syntax described above. The activity will be set when you press the Enter key. Switching to another activity will remove the custom activity from the list.

The activity weight is used by the game to pick an animation with greater chance. Higher values increase the change of the animation being selected. This is typically used to implement a set of idle animations with rarer animations to make an NPC behave a certain way.

The linear movement vector controls the animation's movement in 3D space during the animation. For NPCs this controls the movement speed but not their direction. It is recommended to change only the X value to change the movemement speed. Player movement animations (referred to as gait in the source code) also use only the X value for movement speed.

Models are rendered according to the complete linear movement vector, so changing any of the three values can result in the model appearing to move. Note that entities typically move on their own, so this can have unexpected results. This behavior only affects the visible model, not the underlying entity.

Animations can have events. An event occurs on a specific frame, has an id used to tell the game what to do and a text option to pass as a parameter. Events are commonly used to play sounds, trigger NPC-specific behavior like firing a weapon and control scripted behavior.

Common event ids include:

- 1004 (SCRIPT_EVENT_SOUND): Plays a named sound on the body channel
- 1008 (SCRIPT_EVENT_SOUND_VOICE): Play a named sound on the voice channel
- 1011 (SCRIPT_EVENT_SOUND_VOICE_BODY): Play a named sound with normal attenuation on the body channel
- 1012 (SCRIPT_EVENT_SOUND_VOICE_VOICE): Play a named sound with normal attenuation on the voice channel
- 1013 (SCRIPT_EVENT_SOUND_VOICE_WEAPON): Play a named sound with normal attenuation on the weapon channel
- 5004 (SCRIPT_CLIENT_EVENT_SOUND): Play a named sound on the first free channel

Not all entities and mods support all events.

9.6.6 Body Parts

The Body Parts panel allows to change the enabled submodel for each body part, change the skin and view information about them.

A model has one or more body parts. A body part can have one or more submodels. Entities typically use submodels for things like changing heads, weapons, equipment on the torso and other things.

Submodels have a name taken from the smd filename.

The Body Value contains the value that entities store in the body keyvalue to enable a specific set of submodels.

A model has one or more skins. Each skin is a set of textures applied to the model's meshes. Skins are used to change skin colors, animate eyes and show injuries.

The mesh, vertex and normal counters show how many of each the submodel has.

9.6.7 Textures

The Textures panel allows to preview and modify the model's textures.

Texture scale, the mesh view and the colormap controls are for preview purposes only and do not affect the model itself.

The texture flags control in-game rendering behavior. Some flags are mutually exclusive and cannot be enabled at the same time.

The Fullbright flag is only supported by Svengine and has no effect in vanilla Half-Life.

Textures can be imported and exported. The Import All and Export All buttons operate on a directory basis.

Import All Textures scans the given directory for files matching the name of textures in the model and imports those. Export All Textures exports all textures with the name they use in the model.

Export UV Map allows the previewed UV map to be exported. Because most textures have a low resolution the Export UV Image dialog allows the image to be scaled up to increase the quality of the image.

The colormap controls allow the texture's top and bottom colors to be modified for real-time display. This only affects the preview, the model is not modified.

Remapping works by using a `low`, `mid` and `high` color table index to replace the hue values with the selected top and bottom colors. The range `low` to `mid` inclusive is replaced with the top color, the range `mid + 1` to `high` inclusive is replaced with the bottom color.

A color table is a list of colors used by the image. Half-Life 1 only supports 8 bit indexed images which have a color table containing 256 colors. This means color indices range from 0 to 255.

Color remapping only works if the texture has a specific name:

- If the name has the format `remap<character>_<low>_<mid>_<high>.bmp` then `low`, `mid` and `high` are extracted from the texture name. `character` can be any ASCII character, but it is recommended to use the established convention that uses the numbers `1-9` and the letters `A-Z`. Avoid using unprintable characters and characters already used as delimiters in the rest of the string (underscores, dots and minus/dash signs).
- If the name has the format `remapC_<low>_<mid>.bmp` (C can be lowercase c as well) then only `low` and `mid` are extracted. This is equivalent to the above method with the `high` value set to `000`. This format appears to exist to fix an incorrect use of the above format and should be avoided.
- If the name is `DM_Base.bmp` then `low` is set to **160**, `mid` is set to **191** and `high` is set to **223**.

Color table indices must always contain 3 numbers, for example an index of **10** is **010**. To work properly `mid` must be greater than `low`. `high` can be **0** to disable bottom color remapping, otherwise it must be greater than `mid + 1`.

A special case exists when remapping the first color index to the bottom color. To do this you must use the following values: `remap1_000_-01_255.bmp`. The `high` value can be any color index greater than 0. This is needed because the engine will remap the bottom color starting at `mid + 1` so you must adjust the starting index accordingly. You should not use negative indices in any other case.

Examples:

- `remap1_000_127_255.bmp`: Remap the first half of the color table according to the top color and the second half according to the bottom color.
- `remap1_000_127_000.bmp`: Remap the first half according to the top color. The bottom color is not used.
- `remap1_128_127_255.bmp`: Remap the second half according to the bottom color. The top color is not used. Note that the `low` value is greater than `mid`.
- `remap1_000_255_000.bmp`: Remap the all colors according to the top color. The bottom color is not used.
- `remap1_000_-01_255.bmp`: Remap the all colors according to the bottom color. The top color is not used.
- `remap1_000_-01_000.bmp`: Colors are not remapped at all. Note the use of a negative `mid` index to prevent the first color from being remapped to the top color. This is not particularly useful since the same can be accomplished by not using texture remapping at all.

9.6.8 Model Data

The Model Data panel allows to modify the model's eye position, bounding and clipping boxes as well as the model flags.

9.6.9 Bones

The Bones panel allows to preview and modify the model's bones.

A bone's position, position scale, rotation and rotation scale is the data used calculate the bone's position for animation. This data is usually too specific to be modified directly.

9.6.10 Bone Controllers

The Bone Controllers panel allows to preview and modify the model's bone controllers.

The controller value is only used to preview the controller and does not change the model itself.

Note that all mouth controllers use the same controller value.

9.6.11 Attachments

The Attachments panel allows to preview and modify the model's attachments.

Attachments usually do not have names as this is a feature that was not used by original models, but it is an option and can be used to refer to attachments by name in-game.

9.6.12 Hitboxes

The Hitboxes panel allows to preview and modify the model's hitboxes.

The hitgroup is often entity-specific, refer to the game's source code and/or documentation for more information about which hitgroups are used.

9.6.13 Transformation

The Transformation panel allows to modify the model's data. The user can move the model, rotate it around it's center-most root bone and scale part or all of the model.

Note that rotating models is not a perfect process. Some animations assume the model is facing a certain direction and will not work properly when rotated. Rotate the model in a model editor like Blender for best results.