



J.A.C.K.

Reference Manual

Version 1.1

November 24, 2016



Copyright © Crystice Softworks
2013 — 2016

Introduction

J.A.C.K. (*Just Another Creation Kit*; previously known as **Jackhammer**) is a brand new level editor for games with a quake-style BSP architecture. The aim is to develop a convenient cross-platform tool capable of incorporating the best features of existing editors, such as **Valve Hammer Editor**, **Q3Radiant** and others. Despite Quake and Half-Life having been released a long time ago, the large communities have arisen around, still developing modifications and games on their bases. However the existing editors suffer from fundamental disadvantages their users are well familiar with. **J.A.C.K.** does aspire to be the universal level design tool for classic games. But not only the classics! The editor shall become a key development tool for the Volatile engine, so that is why its second name is **Volatile Development Kit**.

J.A.C.K. is being developed since August 2013. Despite not all the ideas being implemented and not all the functions being completely error-free, you are already able to download fully functional version of the editor, install and evaluate **J.A.C.K.** in action. Please don't forget that any complicated software may contain some issues. We are interested in a vast testing of the editor and grateful in advance for your comments and suggestions!

We are pleased to see you amongst the **J.A.C.K.** users!

J.A.C.K. Steam Store page:

<http://store.steampowered.com/app/496450>

Online information about **J.A.C.K.** is available from:

<http://jack.hlfx.ru>

Def2fgd tool:

<https://bitbucket.org/FreeSlave/def2fgd>

Copyright Notice: some parts of the manual were taken from the official Worldcraft 3.x CHM manual; copyright © 1996-2000 Valve LLC, all rights reserved. The appropriate modifications were introduced in some places, to handle the differences between Worldcraft/VHE and J.A.C.K.

Contents

1. BASIC INFORMATION	6
1.1. KEY FEATURES	6
1.2. HARDWARE AND SOFTWARE REQUIREMENTS	10
1.3. CONTACTING THE AUTHORS.....	10
2. SETUP GUIDE	11
2.1. GENERAL SETTINGS	11
2.2. GAME CONFIGURATIONS	11
<i>Automatic Setup</i>	11
<i>Manual Setup</i>	11
3. LEVEL DESIGN BASICS	16
3.1. INTRODUCTION TO EDITING	16
<i>Solids: the Foundation of 3D Design</i>	16
<i>Entities</i>	16
<i>Entity Types</i>	16
<i>Putting it All Together</i>	16
3.2. THE 3D AND 2D VIEWS.....	17
<i>Using Cameras</i>	17
<i>Camera Placement</i>	17
<i>Keyboard Shortcuts</i>	18
3.3. CREATING SOLIDS.....	18
<i>Creating Solids from Scratch</i>	18
<i>Creating Solids by Cloning</i>	19
3.4. TEXTURING SOLIDS.....	19
<i>Texture Browser</i>	20
<i>Texture Application Mode</i>	21
3.5. RESHAPING SOLIDS.....	22
<i>Vertex Manipulation</i>	22
<i>Vertices and Edges</i>	23
<i>Face Splitting</i>	24
<i>Vertex Welding</i>	24
<i>Vertex Scaling</i>	24
<i>Clipping Tool</i>	25
<i>Brush Merging</i>	26
<i>Carving</i>	26
3.6. CREATING ENTITIES.....	27
<i>Point Entity Placement</i>	28
<i>Solid Entity Placement</i>	28
<i>Names and Targets</i>	28

3.7. USING THE ARCH TOOL	29
3.8. GROUPING AND VISGROUPING	30
<i>Grouping</i>	30
<i>VisGroups</i>	30
3.9. COMPILING AND RUNNING YOUR LEVEL	32
4. ADVANCED TOPICS.....	34
4.1. WORKING WITH MODELS	34
4.2. PATH TOOL	35
4.3. COMPILING IN EXPERT MODE	36
4.4. CREATING CURVES (QUAKE III)	36
4.5. EFFECTS EDITOR (QUAKE III)	36
5. REFERENCE.....	38
5.1. MENUS	38
<i>File</i>	38
<i>Edit</i>	38
<i>Map</i>	39
<i>View</i>	40
<i>Tools</i>	41
<i>Curve</i>	44
<i>Plugins</i>	44
<i>Window</i>	44
<i>Help</i>	45
5.2. DIALOGS	45
<i>Automatic Configuration</i>	45
<i>Check for Problems</i>	46
<i>Configure J.A.C.K.</i>	48
<i>Default Room Properties</i>	56
<i>Entity Help</i>	56
<i>Entity Report</i>	57
<i>Go to Brush</i>	59
<i>Go to Coordinates</i>	59
<i>Hollow</i>	59
<i>Map Information</i>	60
<i>Object Properties</i>	61
<i>Paste Special</i>	62
<i>Replace Textures</i>	63
<i>Surface Properties</i>	64
<i>Synchronization with SteamCloud</i>	68
<i>Transformation</i>	69
5.3. TOOLBARS	69
<i>Map Tools</i>	69

<i>Map File</i>	72
<i>Map View</i>	72
<i>Map Transform</i>	72
<i>Map Operations</i>	73
<i>Map Render</i>	74
<i>Object Bar</i>	75
<i>Texture Bar</i>	76
<i>Filter Control</i>	77
<i>Status Bar</i>	78
5.4. HOTKEY REFERENCE	78
6. F.A.Q.	83
6.1. GENERAL QUESTIONS	83
<i>Quake II/III textures are not displayed — what's wrong?</i>	83
<i>Why do I experience strange errors importing MAP format?</i>	83
<i>Where are configuration files located?</i>	83
<i>Sound playback in Linux doesn't work. What's wrong?</i>	83
6.2. HALF-LIFE QUESTIONS	84
<i>Which format is better to store levels in — JMF or RMF?</i>	84
6.3. QUAKE/HEXEN II QUESTIONS	84
<i>There are no textures, where can I get them?</i>	84
<i>Models are not animated. Is that intent?</i>	84
<i>I get an error “Invalid argument” during a compile process. What's wrong?</i>	84
6.4. QUAKE II QUESTIONS	84
<i>Models are not animated. Is that intent?</i>	84
6.5. QUAKE III QUESTIONS	84
<i>How to create a curve?</i>	84
<i>What is the difference between q3map and q3map2?</i>	85
<i>Why do some shaders issue loading warnings?</i>	85
7. EXTERNAL REFERENCES	86
7.1. OFFICIAL WEB PAGES.....	86
7.2. MAP COMPILERS.....	86
7.3. ADDITIONAL GAME/ENGINE/MOD SUPPORT.....	86

1. Basic Information

At present, J.A.C.K. supports the following base games/engines: **Half-Life**; **Quake**; **Hexen II**; **Quake II**; **Quake III** (including many idTech3-based games, e.g. *Tremulous*).

The editor also supports most of the official and unofficial modifications and add-ons to the games listed. For example, *Half-Life* support includes, but is not limited to *Counter-Strike*, *Team Fortress Classic*, *Deathmatch Classic*, *Gunman Chronicles*, *Sven Coop*, etc. However, the default J.A.C.K. installation package does not provide game definition files (FGD) for most modifications and add-ons. Such files describe game objects and should be written by the user in the same way as default games' definitions are written. Some of game definition files compatible with J.A.C.K. are publicly available on the Web, or may be created from DEF/ENT files (game definitions of Radiant-family editors) using **def2fgd** tool:

<https://bitbucket.org/FreeSlave/def2fgd>

1.1. Key Features

A list of the most prominent (as compared to VHE and/or Q3Radiant) features goes here. The list of new abilities is far from being complete; therefore don't hesitate to look through the habitual menus and dialogs thoroughly!

- **Cross-Platform Editing:** the editor is being developed on a cross-platform Qt framework and is available both for Windows and Linux users. We do our best to provide the same level of comfort to Linux users, revealing them finally a convenient level editor!
- **Large memory addressing support:** 64-bit version on the editor overcomes the 2 Gb memory limit. Now you can store more history items and undo more recent actions.
- **Extensibility:** the editor supports plugins to extend its abilities. Plugins can be very versatile: from new brush primitives, texture and model formats to new game configurations with their own import and export formats. Plugins can share the functionality with each-other: so that WAD3 texture format used by Half-Life can be easily "plugged" into Quake and used along with its "native" WAD2 format, certainly with compilers that also support it.
- **Game Formats Support:** because of flexible plugin system the editor supports many formats of game resources and correctly displays them. They are models of different formats (Quake MDL, Half-Life MDL, Quake2 MD2), sprites (Quake SPR, Half-Life SPR, Quake2 SP2), map sources (MAP, RMF). There is an ability of import and export in RMF format preserving most of its specific features (object colors, grouping, visgroups, etc.), so that J.A.C.K. can be used along with Valve Hammer Editor (VHE).

- **Archive Support:** J.A.C.K. can load models and sprites from game archives (PAK, PK3). This is useful if you open other's map which uses models, and your game resources are not unpacked.
- **Color Schemes:** the editor's look-and-feel can be easily customized using the color schemes, both predefined and user-created. Almost all colors in viewports are customizable — from a background color to a color of Vertex Tool bullet's frame. Besides a general color of the application's window can be altered. In addition one can define his own presets in a configuration file.
- **Real-time Texture Effects:** transparency, texture animation and scrolling support promotes a higher grade of perception of a level under development. Now one can estimate translucency of windows not only in a game, but also in the editor. You will be able to see your Quake III shader scripts almost the same as in the game! Along with that, special rendermodes of textures in Half-Life models are supported.
- **Improved Decal Rendition:** you can preview Half-Life decals (colors and transparency) in a 3D-View just the same as in the game.
- **Embedded Shader Editor:** now the creation process of shaders for Quake III and other games based on its engine becomes delightful. You can not only preview effect changes in real time, but also utilize such convenient features, as automatic source code formatting, syntax highlighting and keyword auto-completion.
- **Large Map Support:** the editor enables working with levels of large size, up to 262144 x 262144 units, now it is simply set up in the Preferences. Now one can easily crate levels for mods with a support of large outdoors.
- **Multiple VisGroups per Object:** hence an object can exist in several visgroups. In J.A.C.K., VisGroups are never created automatically upon hiding a level fragment: hide/show mechanism doesn't rely on grouping anymore. However VisGroups allow hiding certain logical groups of objects along with color coding, in other words, acting as a regulation tool.
- **Textures Bound to Game Configurations:** one more important innovation is texture binding to game configurations instead of a global texture list. This feature allows, for example, address different WAD files in different mods and work with them independently.
- **Loading Resources on Demand:** to speed things up, the editor precaches the resources on startup. However within a considerable amount of game configurations this process may take a long time. To deal with this issue, there is an option to precache only the resources needed in the immediate work. Working with a Half-Life level, the editor precaches only models and textures for this game, saving system's memory. Upon loading a Quake level, J.A.C.K. will add its resources to the load.
- **Dynamic Sky Rendition:** the editor renders skyboxes in the same way games do. In addition, there are layered clouds in Quake, rotating skybox in Quake II, and shader effects in Quake III supported.

To facilitate the choice of a certain sky in map properties, a drop-down combo with icons was added.

- **Curved surfaces:** the editor supports curved surfaces (patches) used in games based on Quake III engine, and also contains instruments making work with them easier: rotating around a control point, randomizing control points, and other. It is possible to create such primitives as Exact Cone, Exact Cylinder, Exact Sphere — correct round variants of cones, columns and spheres.
- **Path Tool:** a new instrument considerably facilitates the process of creating paths for trains, monsters and other objects. You can copy and paste paths in a map, join and split them, invert and convert to linked set of entities.
- **UV-Lock:** the new feature emulates 3D model editor functionality where texture coordinates are bound to vertices. Although such emulation is not completely possible, the editor sometimes allows to achieve previously almost impossible ways to texture faces, e.g. skewing a texture along with a brush side.
- **New Texture Application Modes:** "NULL to Selected" (applying NULL or caulk texture to selection), "NULL to Unselected" (same as the previous mode, but texture is applied to the other brush faces; handy fast removal of back faces), "Apply (texture + values + axes)" (copying texture axes information, simplifying texture application to complex objects and landscapes, especially when combined with "Align to View" mode).
- **Triangulation:** a special command enables triangulation of non-planar faces that frequently arise during vertex manipulation. This helps to get rid of many "Invalid Solid Structure" errors, and to facilitate creation of curved columns and other complex geometry using vertex rotation tool. Simply triangulate your complex stuff after you're done. This command, along with others, is added to a new context menu in Vertex Manipulation mode.
- **Additional Instruments:** along with Worldcraft/VHE instruments one got used to, the editor supports some specific features: either original or adopted from Q3Radiant's ideology. For example, there are such new functions, as brush merging, scaling texture lock, validity restrictions in Vertex Manipulation mode, model rendition in 2D viewports (including the translation/rotation preview), and scaling models with Alt key pressed. On a new level creation, the editor prompts for a basic room, which is very convenient for validation of a game configuration or, perhaps, testing specific entities.
- **Compiling in Editor:** now it is much more convenient to compile maps in the editor, because the compile process doesn't block it. You can continue editing the map while the long compilation takes place. You can also terminate it at any time simply by closing the Process Window.

- **Automatic Selection in 3D:** you can select multiple objects by pressing mouse button in a 3D-View and then dragging a cursor. This mode is convenient for quick selection of lots of nearby objects (e.g. landscape brushes), when clicking can become annoying.
- **Snap to Grid Revised:** subsequent calls to it change anchor point, looping through all eight bounding box points (previously only the lower bounding box point was used as an anchor). In Vertex Manipulation mode this instrument snaps using selected vertices, thus enabling alignment of the whole object that ensures the certain vertex on the grid. In addition, VHE-style snapping to grid during movement of the selection was added.
- **Keep Internal Connections:** option to preserve internal connections (targetname-target links) in entities during copy-paste and cloning (i.e. dragging with Shift pressed) operations.
- **Model View:** you can specify an external model viewer (e.g. HLMV) and open any model simply by a context menu command. Also it is now possible to reload a model from disk without restarting the editor.
- **Extended FGD Format:** the file format describing game entities was extended for the specific needs of J.A.C.K. For example, the new format enables displaying angles of entities using arrows, rendition of light sources as octahedrons, and introduces a help system for entities and their parameters. You can extend your own FGD per existing samples (e.g., "halflife.fgd" distributed with the editor).
- **Autosave:** you can setup automatic save function to protect your project from editor crashes or own design errors.
- **Incremental Save:** a new version saving command automatically adds version number to the file name. Such behavior is familiar to 3DSMax users; it enables easy creation of checkpoints during prolonged project development.
- **Multilingual Support:** switch to your native language when using the editor! Now there is no need to consult a dictionary to find out what either menu command does, or to understand a reason for an error. Presently there are English, Russian, Ukrainian, Byelorussian and Polish languages supported. Languages can even be switched dynamically without restarting the editor.
- **Miscellaneous:** the editor supports precise selection of models in 3D viewport, loading and rendition of portal files (*.prt) and hull files (*.p0-p3), assigning a random yaw to entities, multiple texture fitting, texture replacement with scaling, context menus for entity creation and many other pleasant little things, and there is no good to enumerate them all. Just install the editor and look at your own!

1.2. Hardware and software requirements

J.A.C.K. operates in graphics mode that requires an OpenGL-capable graphics accelerator with up-to-date drivers. However since the program uses mostly a basic OpenGL 1.1 feature set that is usually available even without graphics accelerator through software emulation, it should work even with integrated video solutions. But in order to achieve satisfactory performance, a video accelerator is required (particularly, for texture compression and automatic mipmap generation support an OpenGL 1.4-compliant video card is required).

J.A.C.K. is designed to run at a minimum screen resolution of 1024 x 768 pixels; for some dialogs, a vertical screen resolution of 960 pixels or larger is desirable (otherwise the window may not fit the screen).

The amount of system memory does not impact general performance, but high numbers of undo levels may require a large amount of RAM (2+ Gb). Please adjust the number of undo levels appropriately; otherwise the behavior of the editor may be undefined.

The speed of hard disk drive can be essential for texture loading; therefore a fast drive (for example, SSD drive) is recommended but not required.

J.A.C.K. has been compiled to run on Microsoft Windows® Vista, Microsoft Windows® 7, Microsoft Windows® 8 and newer; UNIX version has been compiled to run on Linux with X-Server (for example, Debian and Ubuntu).

1.3. Contacting the authors

The primary developer of J.A.C.K. is Crystice Softworks. The following persons are acknowledged for valuable contributions and beta testing of the editor's functionality, usability and feature set: DarkEssence, FreeSlave, GNU/Hurt, Phantomas, Skaarj, Scrama, H-3D, Zheka. Localizations were created by XaeroX (russian), Flash (ukrainian), George (byelorussian), Val (polish). We are very interested in and grateful for any user comments and reports of program faults and inaccuracies. If you have any suggestions, bug reports, or general comments about J.A.C.K., please send them to us at support@hlfx.ru.

The official website is <http://jack.hlfx.ru>.

Steam Store page is <http://store.steampowered.com/app/496450>.

2. Setup Guide

2.1. General Settings

The first thing you should do after installing J.A.C.K. is configure its settings. The settings dialog can be reached by selecting Options from the Tools menu, or by pressing **F2**. There are five different tabs in the Configure J.A.C.K. dialog:

1. General;
2. 2D Views;
3. 3D Views;
4. Game Profiles;
5. Misc.

You can leave most of these options at their default settings for now. Once you are familiar with J.A.C.K., it is recommended you play around with these settings until you've got things set like you prefer. More information can be found in the [Configure J.A.C.K.](#) dialog description.

The only essential tab is **Game Profiles**, where game configurations are created and edited (see below).

2.2. Game Configurations

If you want to use J.A.C.K. for game level creation and editing (sure you do; that's why you've installed it), you have to setup game configurations (or profiles) for the games. There are two ways to do this: automatic (Steam version only) and manual.

Automatic Setup

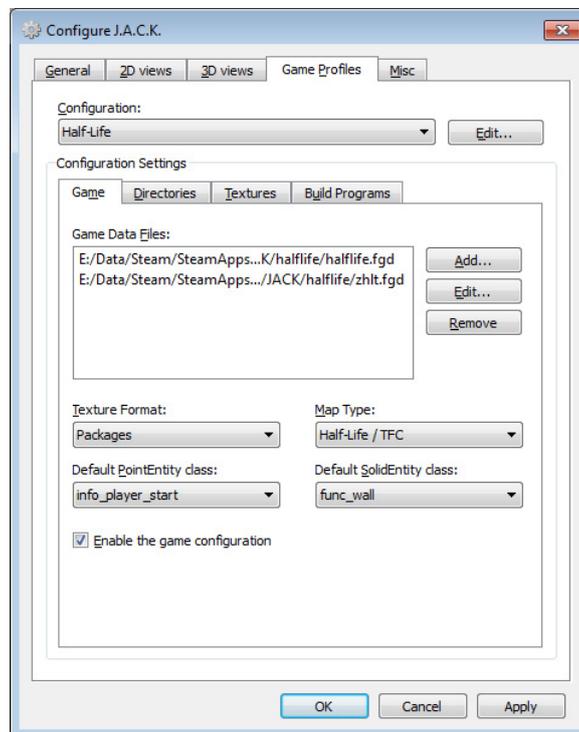
If you have the Steam version of J.A.C.K., you can enjoy the advantages of automatic setup for the games supported. Press **Ctrl+F2** to open **Automatic Configuration** dialog, browse for games' executables and press **OK** to create configurations for them. Please refer to [Automatic Configuration](#) dialog description for the details.

Manual Setup

To create a game configuration from scratch, you should follow a number of steps. Let's create a game profile for Half-Life for Windows operating system. Linux setup is almost the same; simply ignore the executable extension (".exe") in the examples provided. We only briefly describe the options available; for the full reference, read [Configure J.A.C.K.](#) chapter for the dialog description.

1. Open **Configure J.A.C.K.** dialog, go to the **Game Profiles** tab, and press the **Edit...** button nearby the **Configuration** list. **Edit Configurations** dialog will appear.
2. Press the **Add** button; **Add a game** dialog will appear. Enter a new name for the game configuration, e.g. "Half-Life". Press **OK** to close the dialog.

3. Press the **Close** button to close the dialog. The newly created game configuration will be selected in the **Configuration** list. Now you can edit its configuration settings.
4. The default tab is **Game** tab. Press the **Add...** button to add game data files: "halflife.fgd" (describes game objects) and "zhlt.fgd" (adds some map compiler-specific stuff). The FGD files are located in the J.A.C.K. directory, in the "halflife" subdirectory. You can also add "halflife-op4.fgd" instead of "halflife.fgd", if you want to create levels for Opposing Force addon.
5. Leave "Packages" as texture format, and "Half-Life / TFC" as map type. You can also select default point and solid entity classes for your convenience ("info_player_start" and "func_wall" re good choices).
6. Leave the checkbox **Enable the game configuration** checked, so it will be active. Finally, the **Game** tab should look like this:

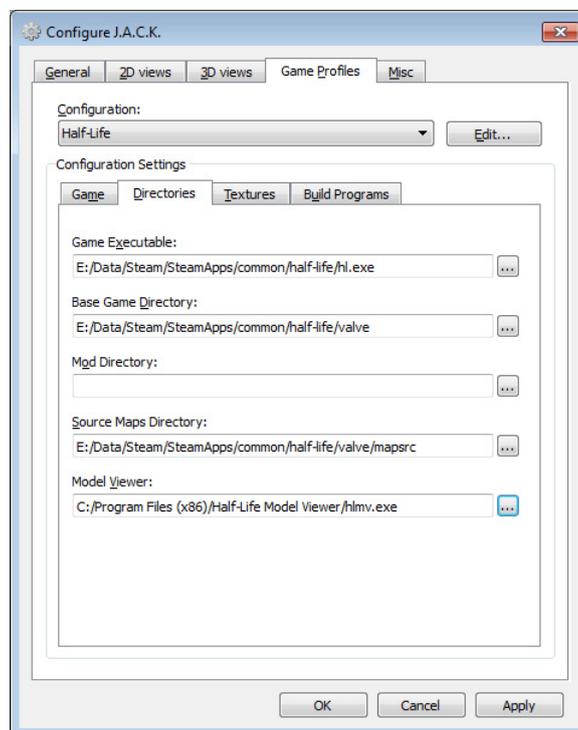


7. Now switch to the **Directories** tab. Here, you should specify executables and directories J.A.C.K. will use for its purposes.
 - **Game executable:** specify the game executable here. If you know the path to the Half-Life executable, type in the path and filename here, otherwise, click on the Browse button and browse to the Half-Life directory. Select hl.exe and click the Open button.
 - **Base Game Directory:** this value should be the base game directory for Half-Life. For Half-Life game configuration, the Base Game Directory should be set to "valve" subdirectory (e.g., "C:/Steam/SteamApps/common/half-life/valve").
 - **Mod Directory:** if you are setting up J.A.C.K. for a mod, the game directory of the mod should go here. Following the above example, if you're setting J.A.C.K. up for Opposing Force, the Mod

Directory value would be "C:/Steam/SteamApps/common/half-life/gearbox". For a normal Half-Life game configuration, the Mod Directory should be left blank.

- **Source Maps Directory:** the value here is the directory where J.A.C.K. will store source files of your maps by default. We recommend creating a "mapsrc" directory in your mod directory. For example, if you've used the example settings from above, open your "C:/Steam/SteamApps/common/half-life/valve" folder and create a "mapsrc" folder, then use "C:/Steam/SteamApps/common/half-life/valve/mapsrc" as a value for your source maps directory.
- **Model Viewer:** if you want to be able to view models with an external viewer (since J.A.C.K. doesn't contain an internal model viewing tool yet; see Working with Models chapter for the details), specify a path to the program here. For example, "C:/Program Files/Half-Life Model Viewer/hlmv.exe" (if you have HLMV installed).

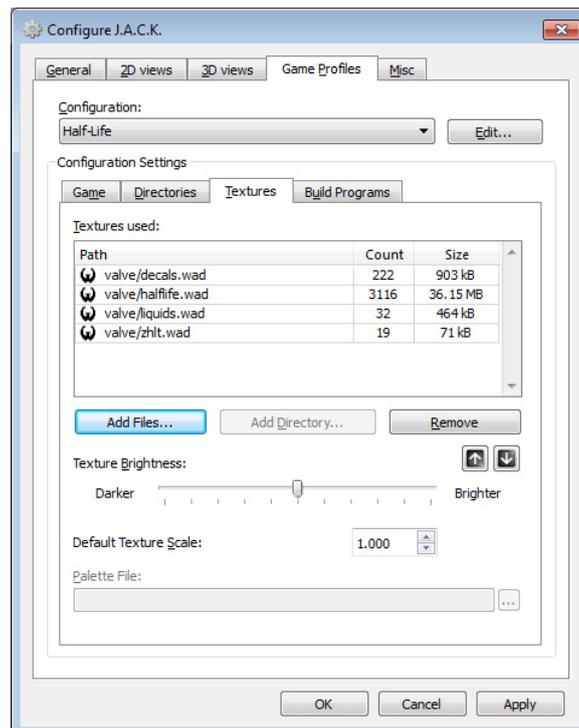
Finally, the **Directories** tab should look like this:



8. Switch to the **Textures** tab. Here, you should specify textures for this game configuration. Unlike Valve Hammer Editor/Worldcraft, J.A.C.K. uses independent texture sets for game configurations; although you can use the same textures in different configurations.
9. Press **Add Files...** and select a set of WAD files (Half-Life texture packages) to use. The recommended initial set is: "decals.wad", "halflife.wad", "liquids.wad", and "zhlt.wad". If you want to make maps featuring Xen, "xeno.wad" should be also included.

Note: if there is no "zhlt.wad" package in your Half-Life game and/or mod directories, copy it manually from the J.A.C.K. installation; it is located in the J.A.C.K. directory, in the "halflife" subdirectory.

10. You can adjust texture brightness here, depending on your needs. This doesn't affect textures themselves; only the way J.A.C.K. renders them.
 11. Leave the default texture scale 1.0, which is a good choice for low-res Half-Life textures.
 12. Since Half-Life doesn't use a shared texture palette, the **Palette File** field is grayed. However, for Quake and Quake II this field is a must; the corresponding “.pal” files are located in the J.A.C.K. directory, in the "quake" and "quake2" subdirectories.
- Finally, the **Textures** tab should look like this:



13. At last, switch to the **Build Programs** tab. Now, we will specify build programs (i.e. map compilers) that convert our J.A.C.K. map into a playable Half-Life level.

Note: Half-Life compilers (VHLT) are not distributed with the Steam version of J.A.C.K. because of copyright issues. You can grab it from here:

- **Windows, 32-bit:** http://jack.hlfx.ru/pub/vhlt_win32_x86.zip
- **Windows, 64-bit:** http://jack.hlfx.ru/pub/vhlt_win32_x64.zip
- **Linux/SteamOS, 32-bit:** http://jack.hlfx.ru/pub/vhlt_linux_x86.zip
- **Linux/SteamOS, 64-bit:** http://jack.hlfx.ru/pub/vhlt_linux_x64.zip

Or, visit the official VHLT page:

<http://forums.svencoop.com/forumdisplay.php/217-Vluzacn-s-Map-Compile-Tools>

After you downloaded the compilers, unpack the archive to the Half-Life directory (where hl.exe is located); the new "tools" subdirectory containing the executables and other necessary files will be created.

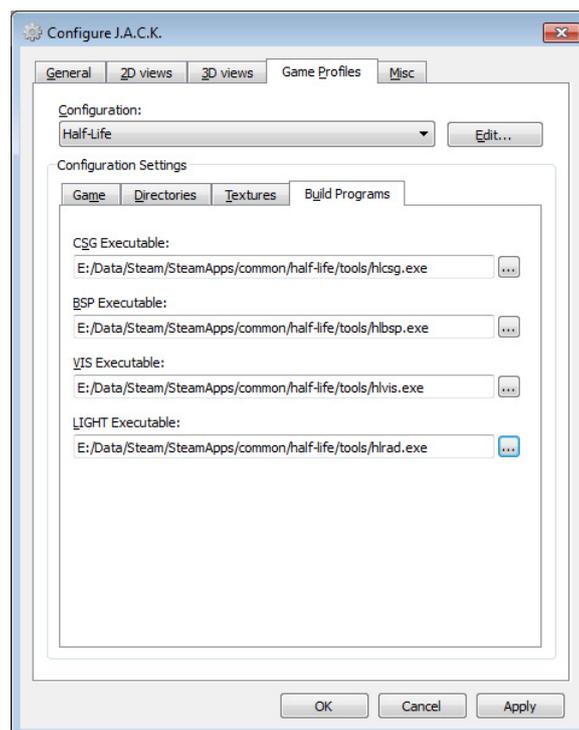
Note: Quake, Quake II and Quake III compilers are distributed with J.A.C.K. under GPL license; they are located in the J.A.C.K. directory, in the "quake", "quake2", and "quake3" subdirectories.

- Specify paths to CSG ("hlcsg.exe"), BSP ("hlbsp.exe"), VIS ("hlvis.exe"), and LIGHT ("hlrad.exe") executables. If you followed the above steps, and assuming directories we used as examples, the CSG executable will be "C:/Steam/SteamApps/common/half-life/tools/hlcsg.exe", the BSP executable will be "C:/Steam/SteamApps/common/half-life/tools/hlbsp.exe", etc.

Note: CSG executable is needed only for Half-Life; in Quake games, its job is done by BSP executable; so you should leave the CSG field blank for Quake, Quake II and/or Quake III.

You can also specify programs' arguments here, right after the path to the executable. This is especially important for such games as Quake III; there is a single map compiler for each compile stage, and the difference is only in arguments. So, you will put "path/to/q3map2.exe" to the BSP executable field, "path/to/q3map2.exe -vis" to the VIS executable field, and "path/to/q3map2.exe -light" to the LIGHT executable field; and that actually works.

Finally, the **Build Programs** tab should look like this:



That's it! Congratulations, now you can create Half-Life levels in J.A.C.K.

3. Level Design Basics

3.1. Introduction to Editing

Solids: the Foundation of 3D Design

Blocks. Wedges. Cylinders. Spikes. They may not sound like much, but these are the basic building blocks of all architecture created in J.A.C.K. You can carve them, clip them, and manipulate them. You can combine these solids (also called **brushes**) to make any shape possible, real or imagined. This is known as constructive solid geometry (CSG) and this is the editing style J.A.C.K. uses.

Once you create a brush, you'll assign to it a texture, which is a pre-existing bitmap image created to make the brush resemble something in the real (or some imagined) world. Examples of textures include bricks, rock faces, and water.

Entities

You say you want more in your game world than inanimate solids? Well then, what you want are entities. Where brushes are "world objects" used to form the basic inanimate structure of your level, entities are the objects that move, have sound, or are interactive. An entity is anything that performs some type of operation or task within your level.

Entity Types

There are two types of entities: point-based and brush-based.

Point-based entities exist only at a certain exact point. Examples include lights, monsters, and players. (Monsters do have an area, but this is defined by the game code and is not modifiable from within the map.) Some point entities are just that: points. For example, the "env_beam" entity, which controls Half-Life's beam effects, uses two point entities as targets; you place the two points and the beam of light runs between them.

Brush-based entities are entities that depend on a brush for their physical presence, like doors, trains, and other moving objects. A trigger is another type of brush-based entity; it requires that you indicate an area or activation field which controls the trigger's operation.

Putting it All Together

Using these simple components, you can create a virtually limitless variety of levels. Whether it's a barren room or a vast, complex world, you'll do it by using solids and textures to create your architecture, then adding lights, monsters, buttons, moving platforms and a host of other entities to bring your creation to life.

Once everything is in place, you will need to compile your level. This is the process that turns your collection of solids and entities into a playable level that you can run in the game. Although the compiling process happens when you think you've finished your level, knowing something about this

process ahead of time can save you many headaches. Be sure you read the appropriate sections of the manual before you start building a level.

3.2. The 3D and 2D Views

Before you can begin creating and placing solids and entities, you need to familiarize yourself with the J.A.C.K. interface. By default, your working area is divided into 4 sections. The first (top, left) window is the 3D View, and the rest are the 2D views.

The 3D window is a dynamic 3D space where you can view your level from any angle. This is critical for checking textures and texture alignment, spotting leaks and just getting a sense of what your finished level will look like.

Using Cameras

To take advantage of the 3D view, you need to be able to place cameras. Cameras determine your vantage point(s) in the 3D view. J.A.C.K. provides you with precise control over the camera movements in your map.

A camera in J.A.C.K. (as displayed in the 2D views) consists of two parts: the eye, and the viewing angle, which is represented by a line extending out from the eye. The length of the line that represents the viewing angle is not important; though it can help you aim the camera exactly at an object.

While it is possible to move a single camera all over the map each time you need to look at something new in the 3D window, it is more convenient to have easy access to multiple cameras placed throughout the map. J.A.C.K. allows you to easily cycle through multiple cameras by pressing PageUp and PageDown.

Camera Placement

Placing cameras in J.A.C.K. is extremely simple. First, switch to Camera mode by pressing **Shift+C**, then hold **Shift** and with the left mouse button, click-drag a line in one of the 2D views. This will create a thin red line with a large dot at one end. The dot is the camera's position, and this is where the 3D camera view will originate. The red line is the camera's viewing angle. You can adjust either end of the line to change the view. Follow the above steps to create as many cameras in your level as you need.

There are a number of options available to you when using multiple cameras. You must have the Camera tool selected to take advantage of these:

- **PageUp**: cycle up to the next camera position
- **PageDn**: cycle down to the last camera position
- **Delete**: delete the current camera position
- **Shift**: hold shift and click-drag with the left mouse button to create a new camera.

While in camera mode, you can adjust the camera position by moving the eye or viewing angle in any of the 2d windows. Hold **Ctrl** to drag both camera and viewing angle points.

Mouse look / NoClip Style Movement

Mouse look movement is designed to be the same as when you are in the game and walking around with "+look" (mouse look) and the NoClip cheat both turned on. It can be enabled or disabled by pressing **Z**.

Moving your mouse around will change the player's direction of focus, while **W** and **S** control forward and backward movement, and **A** and **D** control sided to side (left and right strafing) movement. You can disable this movement style by going into the 3D View options and disabling use of mouse look navigation.

Keyboard Shortcuts

There are a number of keyboard shortcuts that you can use to quickly maneuver through the 3D view without switching to the Camera tool.

- Holding the left mouse button allows you to rotate your angle of view in any direction, while the viewing point remains stationary.
- Holding the right mouse button will allow you to move left, right, up, and down while keeping the viewing angle constant.
- Holding the left and right buttons and moving the mouse causes the view to strafe forward, backward, right, and left. (note that this makes the following behavior redundant).

While holding **Shift**:

- The left mouse button acts the same as above.
- The right mouse button allows you to move forward and backward, as well as from side to side.

3.3. Creating Solids

Creating Solids from Scratch

Creating solids is quite simple. The process always follows the same four basic steps:

1. Select the Block tool by pressing **Shift+B**.
2. In the **New Objects** panel, select **Primitives** in the **Categories** box. In the **Objects** selection box you will have 7 options. Select any one of these objects:
 - block;
 - wedge;
 - cylinder;
 - spike;
 - sphere;
 - rock;

- arch.

The arch primitive is a special case. It is not actually a primitive object, but an **Arch tool** which allows you to quickly and easily make multi-piece arches. For more information on the **Arch tool**, see [Using the Arch Tool](#) chapter.

3. Drag a box in a 2D view, making sure it is the correct size and in the correct position in each of the views.
4. Press **Enter** to create the solid.

Creating Solids by Cloning

Cloning provides a simple way to quickly duplicate an object or group of objects. This includes both brushes and entities, and groups can contain a mixture of both.

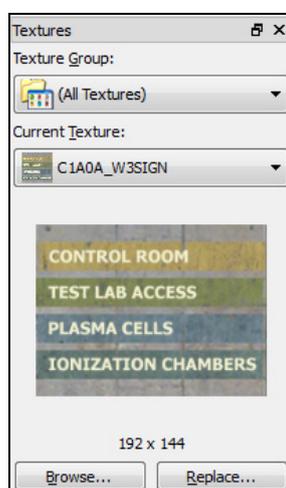
1. Go into Selection mode by pressing **Shift+S**.
2. Select the object (brush, entity, or group of brushes and/or entities) that you would like to clone.
3. Hold **Shift**, and click and drag the selected object to its new position with the left mouse button.
4. While **Shift** is still being held, release the mouse button. The object has now been cloned.

3.4. Texturing Solids

Textures are the designs applied to the surfaces of solids. These textures show what material a given solid is made of (brick, steel, sand, etc.). So, while solids give your level form, textures indicate function, create atmosphere and bring your level to life. Skilled application of textures is one of the most important aspects of creating a good level. J.A.C.K. tools give you easy access to all of the texture properties.

Please note that some games and engines (e.g. Quake III) operate not on textures themselves, but on effects, which are actually one or several textures combined in a way described in a small text script (“effect script”). Please refer to [Effects Editor](#) chapter for the details on how to create, modify and use effects.

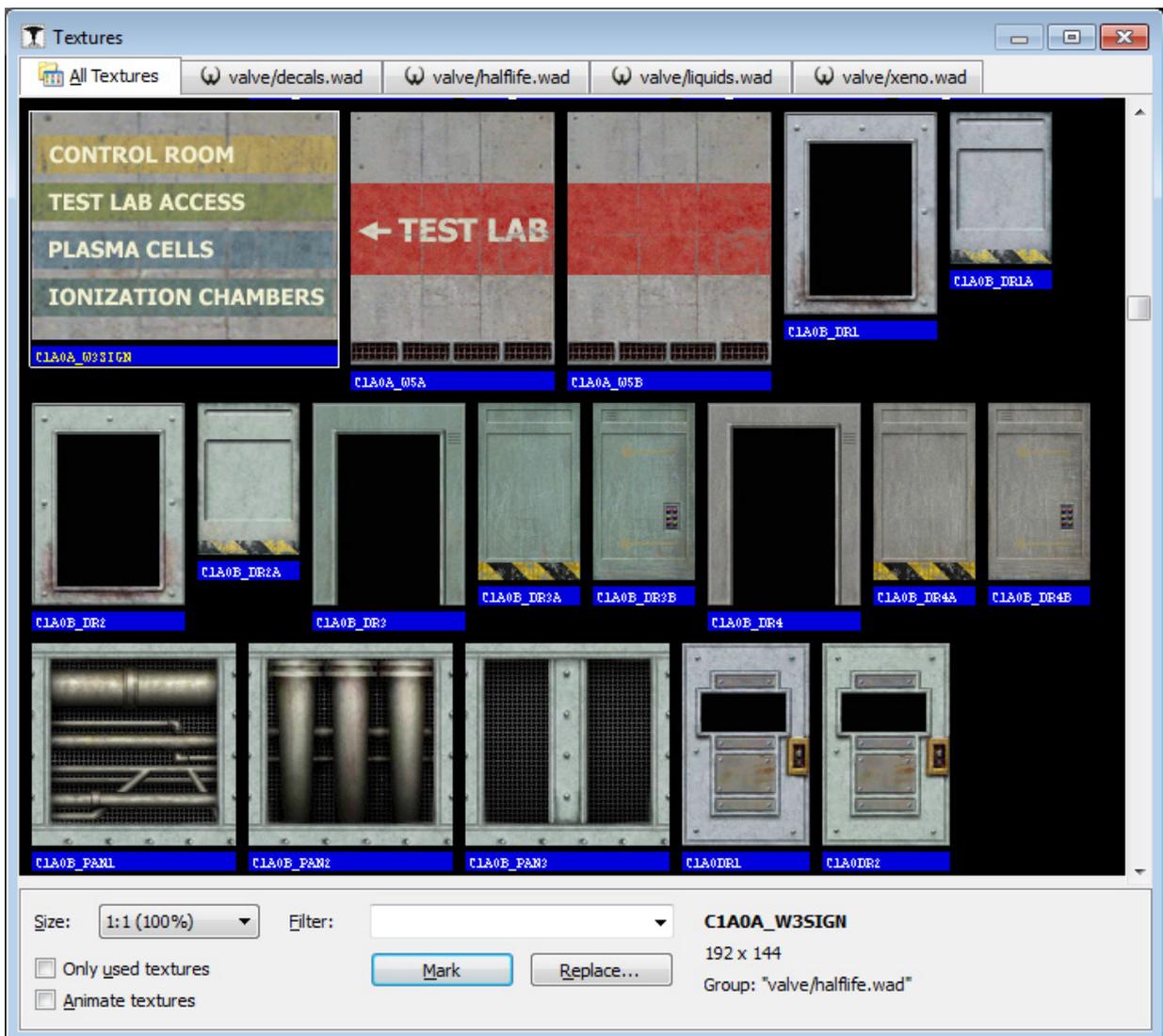
Your first encounter with textures will most likely be through the **Textures** panel.



The **Textures** panel shows the current texture, and lets you select a new texture. The **Current texture** drop-down list shows the last eight textures used; then continues with a list of the remaining textures. Use the **Texture group** scrollbox to view a different subset of textures. You will have a choice between viewing all textures, or viewing individual texture packages (WAD packages, effect script files, texture directories, etc.). You can add or remove texture packages in the Game Profiles, in **Configure** dialog.

Texture Browser

The **Browse...** button opens the texture browser, where you can select from all available textures (as dictated by the **Texture Group** setting).



The **Size** scroll box allows you to set the displayed size of the textures. You can choose from 32x32, 64x64, 128x128, 256x256, 1:1 (full size), 1:2 (half size), 1:4 (quarter size), and 2:1 (double size).

The **Filter** dialog box allows you to type in a search string. Any texture that includes that string as part of its name will be displayed. You can specify more than one filter string. For example, if you

specify the search string “sign 7”, any texture with “sign” and “7” in its name will be displayed. Previous search strings can be accessed by clicking on the down arrow.

The **Only used textures** option tells the browser to display only those textures that have been used in your level. This helps you limit your use of different textures, and avoid using different textures for the same purpose. For games such as Half-Life, textures may be stored in the compiled maps. Reducing the number of textures will decrease the compiled map size. Fewer textures to load also means memory savings while playing the map in the game.

The **Animate textures** option enables texture animations and other special effects, so you can review them immediately.

The **Mark** button allows you to mark (select) all texture faces that contain the selected texture. You can then adjust those faces as required (textures, surface attributes, etc). This is best used in conjunction with the **Surface Properties** dialog. If the normal Apply Textures map tool is used, any brush with one face selected will have the new texture applied to every face of the brush.

The **Replace...** button (also accessible through the Tools menu) will bring up the **Replace Textures** dialog.

Texture Application Mode

You will eventually need to modify a texture in some way. The **Surface Properties** dialog (Texture Application mode; **Shift+A**) allows you to manipulate the texture properties of selected brush faces. To select a brush face, click on it in the 3D view with the left mouse button. To select multiple

faces, hold down the Ctrl key while you do this. You can also select multiple brushes *before* bringing up the **Surface Properties** tool.

Scale (X/Y)

You can modify the scale of a texture to shrink or enlarge it. Numbers less than 1 shrink, and numbers greater than 1 enlarge. You can use negative values to mirror the texture in one or both planes.

Shift (X/Y)

This simply allows you to shift the texture around in the X and/or Y planes.

Rotation

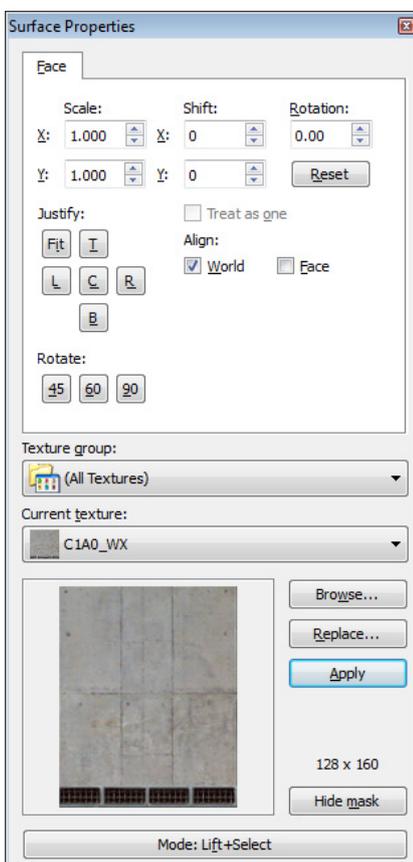
Rotation allows you to rotate the texture on a brush face.

Reset

This button resets scale, shift and rotation to the defaults.

Justify

There are six buttons you can press here: **L** (left), **R** (right), **Fit**



(fit to face), **T** (top), **B** (bottom), and **C** (center on face). These allow you to quickly and easily align a texture on a brush.

Treat as one

If you have multiple faces selected, you can check this option to treat all the faces as one single face.

Align: World and Face

These checkboxes allow you to set the texture alignment style for the selected faces. *World* alignment causes textures to be aligned according to world coordinates (if you're familiar with old versions of Valve Hammer Editor/Worldcraft, this was the standard behavior). *Face* alignment uses the coordinates of the face as a basis for alignment.

Rotate

There are three buttons you can press to quickly rotate the texture by 45, 60, or 90 degrees. Hold Shift when clicking these buttons to rotate in the opposite direction.

Texture Group

You can select which group of textures you want to use (see above).

Current Texture

The name of the current texture is displayed here, along with a thumbnail picture of the texture.

Hide Mask

If this button is toggled on, the selected brush face(s) will not appear as selected (with a reddish hue). This may make it easier to perform some texture manipulations.

Mode

This lets you select what feature you want the left mouse button to be. The available options are Lift + Select, Lift, Select, Apply (texture only), Apply (texture + values), and some others. For more details, refer to [Surface Properties](#) dialog description.

3.5. Reshaping Solids

Once you've created a simple, textured solid, you can use J.A.C.K. tools for refining your work. These tools include vertex manipulation, face splitting, vertex welding and scaling, clip planes, brush merging and carving. Each of these tools lets you customize your solids by modifying the basic shapes you started with. You will find that you can achieve the same or similar effect using different tools; some are just faster and easier than others when it comes to making the precise change you are trying to make. See also [Selection Tool](#) features which are accessed easily and still very effective.

Vertex Manipulation

Vertex manipulation allows you to move individual vertices and edges of a solid, easily creating irregular shapes. This lets you make shapes that are impossible with plane clipping, and extremely

difficult to do with carving. Vertex manipulation is also useful for modifying specific pieces of a group of objects, where resizing would interfere with other objects in the group.

While Vertex Manipulation makes the creation of new complex shapes easy, it also makes the creation of invalid shapes easy. Remember that concave shapes are not valid within J.A.C.K. If you are not sure of what you're doing, enable **solid validity restrictions** mode. In that case J.A.C.K. will block attempts to create invalid solids, although it will be much harder to drag vertices to proper positions.

Vertices and Edges

A vertex (or plural, vertices) is a corner of a solid. Moving one vertex will not effect any other vertex, but it will change the position of any edges associated with it. An edge is a point between two vertices which, when moved, will change the position of its two associated vertices, as well as their associated edges (no other vertices are effected besides the first two).

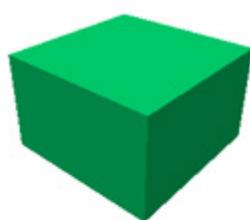


Fig. 1

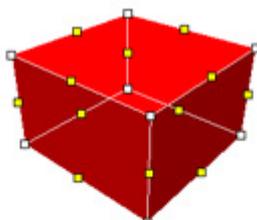


Fig. 2

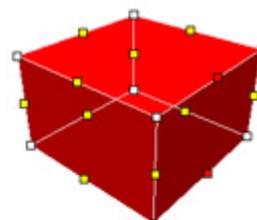


Fig. 3

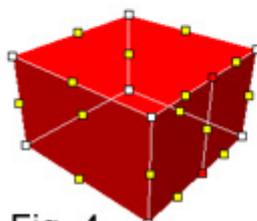


Fig. 4

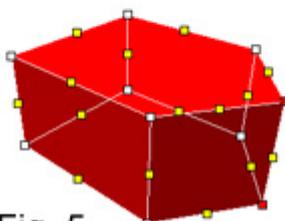


Fig. 5

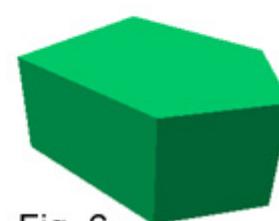


Fig. 6

1. Select the object(s) you would like to manipulate (Fig. 1).
2. Select the **Vertex Manipulation** (VM; **Shift+V**) tool from the MapTools toolbar. The selected solid will turn into a wireframe-type solid with partial shading (Fig. 2). Vertices will be marked as white dots, and edges as yellow dots.

Note: You can keep pressing the VM button, or press **Shift+V**, to cycle through the 3 VM modes. They are: vertices and edges (default), vertices only, and edges only.

3. Click on the vertices/edges you want to move. They will be highlighted (Fig. 3). Note that you can move both vertices and edges. If you don't want to deselect brushes accidentally, hold **Alt** while clicking.

- You can either drag the selected vertices or edges with the mouse, or move them in the selected 2D view using the arrow keys (Fig. 5).

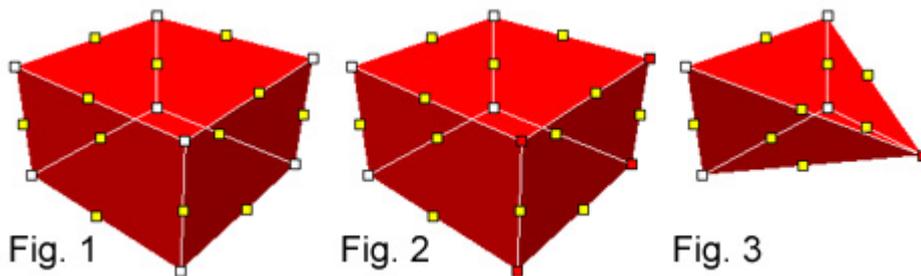
Tip: To undo anything done in VM mode, you will have to first exit VM mode (Fig. 6).

Face Splitting

Face splitting allows you to add additional faces to a brush. This is a useful tool when you want to add complexity to an object in your level. Simply select two opposing edges (Fig. 3) and press **Ctrl+F**. Alternatively, you may right-click to invoke a context menu, and select the “Split Face” command from the menu. Anyway, this will turn the two edges into vertices, and place a new edge between them (Fig. 4).

Vertex Welding

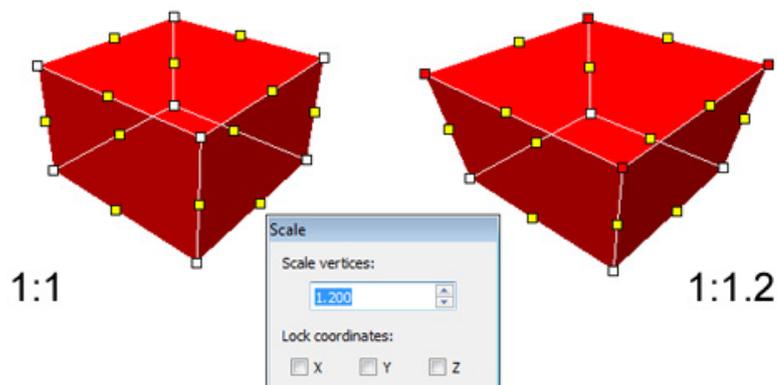
Vertex welding moves selected vertices to a single point (an average point, actually a “center of mass”), and they will be subsequently merged (upon exiting VM mode) thus reducing complexity of the solid.



In the VM mode (Fig. 1), select vertices you want to weld together (Fig. 2) and press **Ctrl+Alt+W** (or right-click to invoke a context menu, and select the “Weld Vertices” command from the menu). The vertices selected will be joined, transforming, e.g., a box into a spike (Fig. 3).

Vertex Scaling

Vertex scaling allows you to select a number of vertices and change their scale in relation to each other. The process is quite simple.

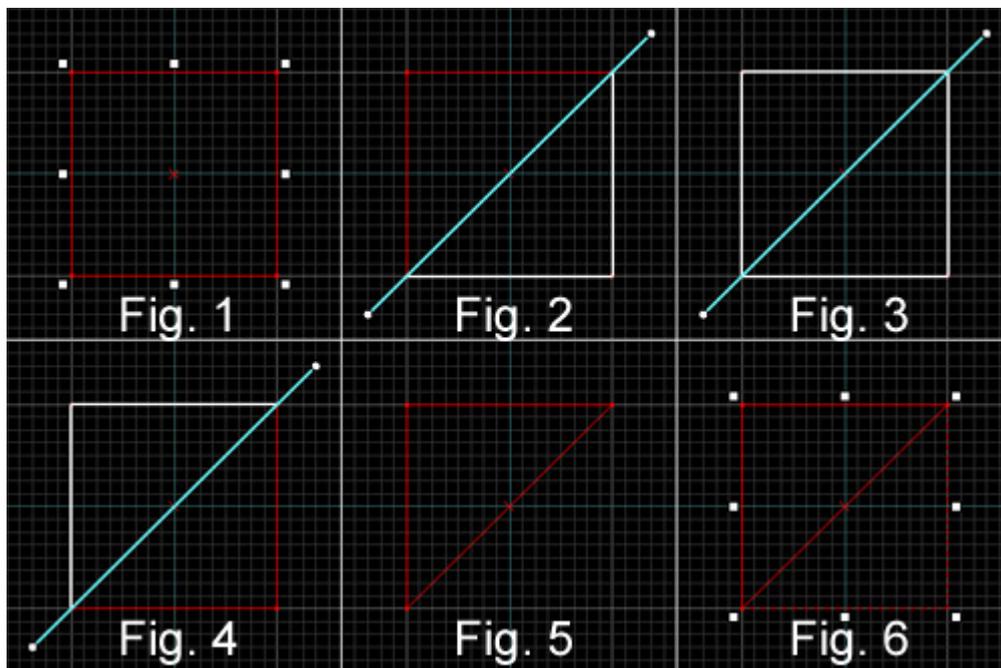


- Select an object.

2. Go into VM mode (**Shift+V**).
3. Select a number of vertices. In the example picture below, all of the vertices on the top of a block have been selected.
4. Press **Alt+E** to bring up the **Scale** dialog (or right-click to invoke a context menu, and select the “Scale Vertices” command from the menu). You can press the up or down arrows to move in 0.1 increments, or enter a scale value directly in the text area. The scale changes will automatically be reflected on the selected object. Use checkboxes to prevent scaling along certain axes, if needed.
5. When you are done with the scaling, press Esc to close the **Scale** dialog.

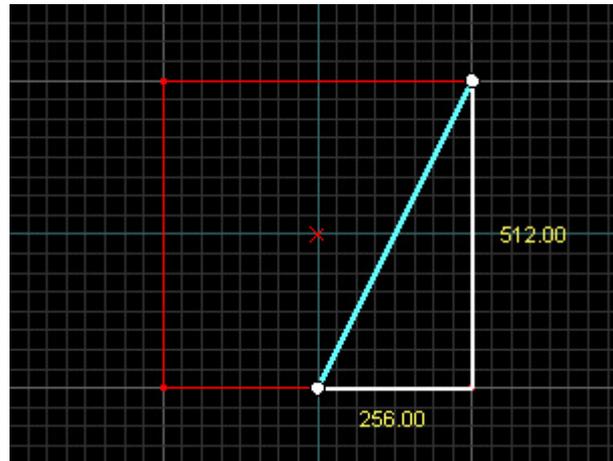
Clipping Tool

Clip planes let you make a precise cut in a solid, dividing it into two pieces. You then have the option of keeping one or both of the resulting two solids. You may find using Clipping Tool quicker and more efficient than carving or vertex manipulation.



1. Select the object to be clipped (Fig. 1).
2. Select the **Clipping Tool** button on the [Map Tools](#) toolbar (**Shift+X**).
Note: You can cycle through the clip modes by clicking on the button, or by pressing **Shift+X** repeatedly (Fig. 2, 3, 4). You can move both points of the clip line by holding **Ctrl** and dragging one point of it.
3. Drag a line across the selected object (Fig. 2). This is the clip line. The part of the object to be kept will be highlighted in thick white lines. You will also see a color filled clipping plane in a 3D view.
4. When you are satisfied with the resulting object, press **Enter** to perform the clip (Fig. 5, 6).

There is a useful feature to display the size of the solid area of the clipped brush. This can help if you're trying to clip a brush to an exact size. To toggle this on and off, have a 2D view in focus and press the **O** hotkey.



Brush Merging

Brush merging can be useful if you want to make a single solid from a set of solids forming a convex hull. Map compilers of some games (Half-Life, Quake) automatically perform face merging; some (Quake III) don't. In the latter case, careful brush splitting and merging can play an essential role in level design and optimization. Please note that only those brushes forming a convex hull can be merged.

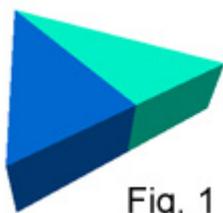


Fig. 1

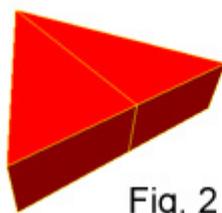


Fig. 2

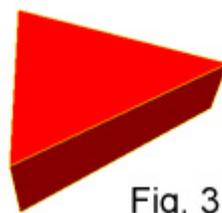


Fig. 3



Fig. 4

To merge brushes, perform the following steps:

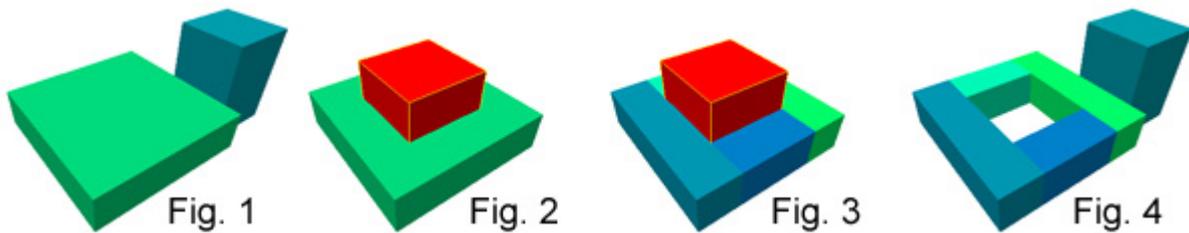
1. Make sure brushes form a convex hull (Fig. 1); concave hulls are not allowed in J.A.C.K. and are treated as invalid solid structures.
2. Select brushes to merge (Fig. 2).
3. Press **Ctrl+Shift+M**, or press the **Merge** button on the [Map Operations](#) toolbar (Fig. 3).
4. If everything is correct, you will get a single solid instead of the set of solids (Fig. 4).

Carving

In J.A.C.K., any solid can be used to carve a volume out of other solids. For example, you can place a solid within the room's wall and tell J.A.C.K. to subtract the solid from the wall — effectively punching a hole right through the wall! This feature is not limited to using cubic shapes as a carving tool: any solid — cylinders, cubes and wedges — can be used to carve other solids.

While carving, especially with a cylinder, it is best to carve the smallest areas possible. For example, if you attempt to create a hole in a large wall using the cylinder, J.A.C.K. will automatically break the wall face into several pieces, something you don't want. Instead, a more efficient way to carve would be to first carve a square hole (the same size as the circle) and leave the square in place. Then you can take the cylinder and carve a hole in the small square. This will eliminate the splintering effect for the wall.

Tip: carving with very complex high-poly primitives (arches, spheres and rocks) is not recommended, because it can produce lots of brush fragments, which may lead to compile errors or in-game glitches.



1. In this step (Fig. 1), you have the object you want to carve into (in green), and the object you will be carving with (in blue).
2. Position the carving object so that it is where you want the hole to be. The carving object should be selected (Fig. 2).
3. Press **Ctrl+Shift+C**, or right click on the selected object and select **Carve** (Fig. 3). You can also press the **Carve** button on the MapOperations toolbar.
4. Either delete the object you used to carve, or use it to fill the hole. For example, use the piece carved out of a doorway as the door. You can also move it away to carve another object (Fig. 4).

3.6. Creating Entities

Entities give life to your level. They are the monsters, doors, switches and lights that turn your static architecture into an interactive environment. Careful use of entities is critical to creating an interesting, unique and fun level for others to play.

Unlike solids, entities are not created within J.A.C.K. Most users will simply choose from a pre-existing set of entities provided with the game you make levels for. Once you have selected the appropriate entity, J.A.C.K. gives you control over its placement and other properties.

There are two types of entities: point-based (point entities) and brush-based (solid entities).

Point entities exist only at a certain exact point. Examples include lights, monsters, and player start points (monsters do have an area, but it is defined by the game code and is not modifiable from within the map). Some point type entities are just that: points. For example, the `env_beam` entity, which controls Half-Life's beam effects, uses two `info_target` entities as targets and the beam of light runs between these two points.

Solid entities are entities that depend on either a brush for their physical presence (like doors, trains, and other moving entities) or an “area” (like triggers, which require an activation field).

Point Entity Placement

Placement of point entities is simple:

1. Press **Shift+E** to go into entity mode..
2. From the **New Objects** toolbar, select the entity you want.

Note: the entity list in the **New Objects** panel is for point-based entities only. For information on brush-based entities, please refer to the following section.

3. Click in a 2D window, then position the entity cursor with the mouse.
4. Pressing **Enter** will create the entity.

You will most likely need to edit the entity properties after you place a point-based entity. Select the entity and press **Alt+Enter** to view the entities properties dialog box. This dialog pops up automatically for brush-based entities.

Solid Entity Placement

Brush-based entities are a bit more complicated than point-based entities, but should not pose any difficulties once you know their basics.

1. Create the object where you would like it. You can use as many brushes as you'd like when building a brush-based entity.
2. Select the entire object.
3. Press the **Ctrl+T** hotkey, which will turn the selected objects into a brush based entity and...
4. ...bring up the entity properties dialog. Select the appropriate entity from the **Class list** box. Once the entity type is selected, you can modify the entity properties as needed.
5. You will notice that brush-based entities appear as a different color in the 2D windows, typically purple.

Bring up the entity properties for a brush-based entity the same way you do for a point-based entity: select it and press **Alt+Enter**.

Names and Targets

Almost each entity may have a name (usually a “targetname” key) and a target (usually a “target” key). This enables scripted connections between entities. For example, there is a light named “light1”, and a func_button with a “target” set to “light1”, this button will toggle the light on use. You can specify names and targets using [Object Properties](#) dialog (**Alt+Enter**).

There is another, simple way to connect two entities:

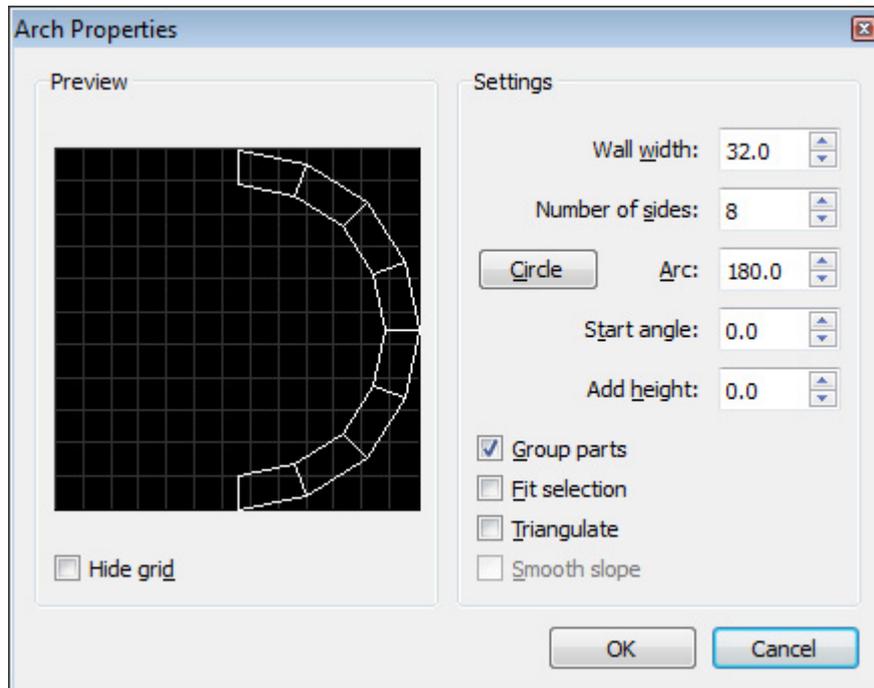
1. Select a first entity which you want to target to some other entity (button, etc.)
2. Select a second entity which you want to be a target of the first entity (light, effect, etc.)

3. Press Ctrl+K, or select the Connect entities command in the Tools menu.

That's it! Now the first entity is connected to the second entity, and will activate it on use.

3.7. Using the Arch Tool

The **Arch Properties** dialog allows you to make complicated curved arches or circular staircases simply and quickly. To invoke it, create a primitive of the type “arch”: this dialog will appear immediately.



There are a number of options you can modify here:

Wall width: this is the width of the individual pieces of the arch.

Number of sides: enter a value between 3 and 100.

Arc: this is the number of degrees the arch will span. You can enter a value here that is between 8 and 360. Press **Circle** button to quickly fill this field with the value of 360.

Start angle: this lets you adjust the angle of the arch pieces. Zero angle means starting from the top (this corresponds to entity angle 90); and unlike entity angle, rotation is clock-wise.

Add height: each successive piece of the arch will be raised this many units. This allows you to make, for example, a curved stairway. In that case, you'd set this value to the height you want each stair to rise.

Group parts: this will create a single group for all the pieces.

Fit selection: this option will fit the pieces to the entire brush space.

Triangulate: arch pieces will be triangle prisms instead of quad prisms; this may be useful for subsequent vertex manipulation.

Smooth slope: if the “add height” value is non-zero, this enables creation of a smooth ramp instead of a stairway.

Hide grid: this is a preview option which disables grid in the preview canvas, and does not affect the arch primitive created.

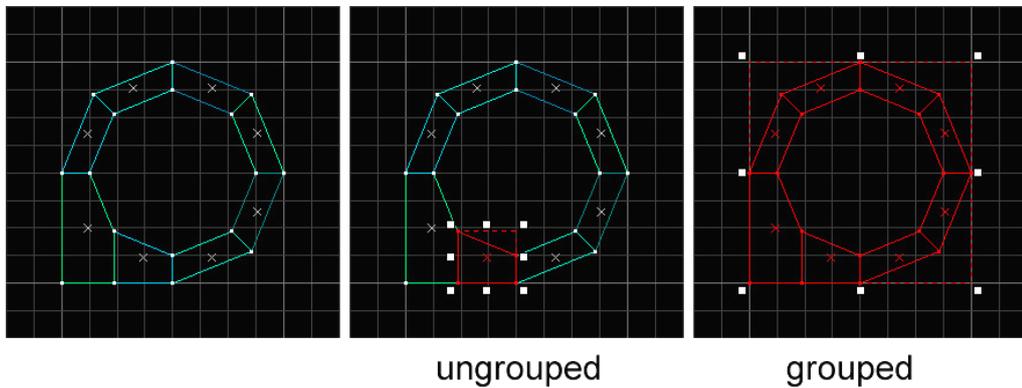
3.8. Grouping and VisGrouping

Even relatively modest J.A.C.K. levels can contain hundreds — even thousands — of objects. The editor offers two ways of keeping track of items in your level: normal grouping and visibility grouping (VisGroups).

VisGroup refers to "visibility group" and has nothing at all to do with the VIS compile tool. VisGroups only affect what you see in J.A.C.K. and have no in-game effect.

Grouping

Normal grouping is used to literally “glue” things together.

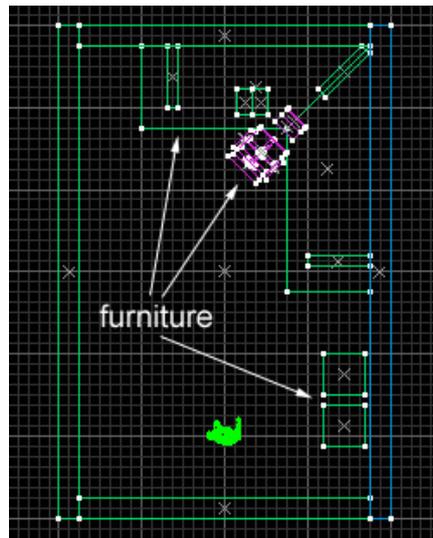


The object shown above is made of eight brushes. If you were to click on a single brush when the object is ungrouped, only that brush would be selected; you'd have to **Ctrl**-click each piece to select the whole thing (or drag a selection box over the whole thing). When the object is grouped, however, you can click on any individual brush to select everything in entire group.

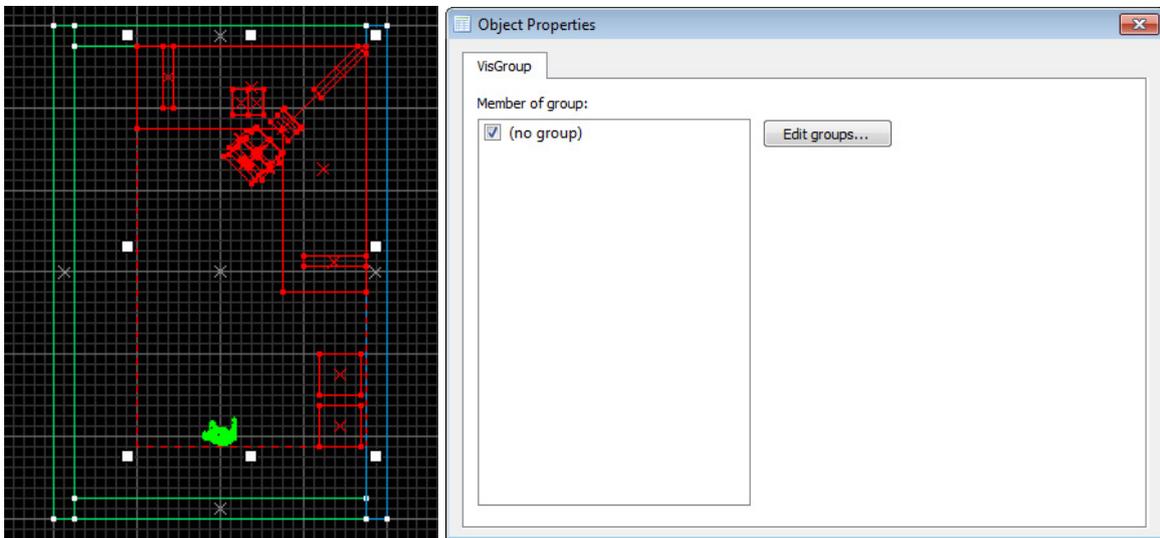
- Brushes do not need to touch in order to be grouped together.
- Groups may contain any combination of brushes, entities and other groups.
- You can work on an individual piece of a group of objects by enabling the ignore groups option.
- Grouped objects can be combined with VisGroups (see below).

VisGroups

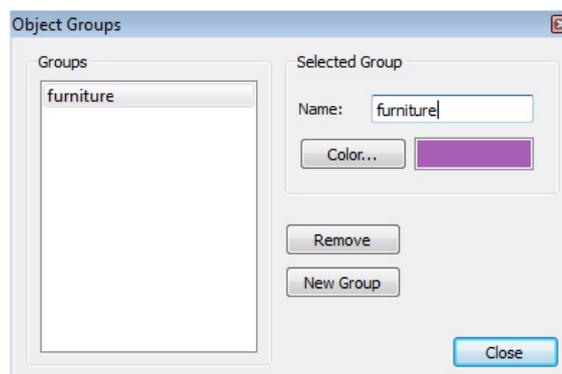
VisGroups stands for visibility groups. VisGroups make managing objects easy by letting you control what you see on your screen. With VisGroups, you can arbitrarily assign objects to a specific group; you can then turn the visibility of that group on and off. Let's say you have the following section in your map:



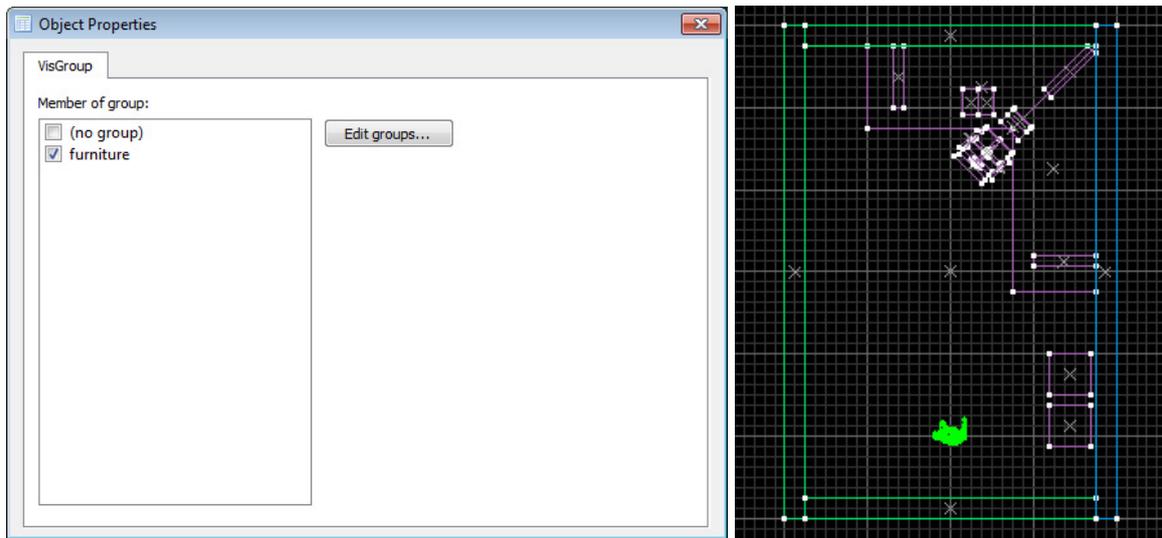
First, select the furniture pieces, and then press **Alt+Enter** to open [Object Properties](#). The **VisGroup** tab should be active.



Click **Edit groups...**, then **New Group**. Enter the group's name (“furniture”). You can also select a representative color for the group. Close the dialog to proceed.



The new group should appear in the **VisGroup** list of objects' properties. Check the box to assign all the selected objects to the “furniture” visgroup. When you deselect the objects, you'll find that all of them are now rendered in the VisGroup's color.



The new VisGroup will appear in the **Filter Control** panel, with the number of objects it contains, in brackets, if any. To change the name of the VisGroup, select the VisGroup, then press the **Edit** button. You will be able to change the name (along with the color) to whatever you like. You can also delete the VisGroup by pressing the **Remove** button. Deleting the VisGroup does not actually delete the group's objects; it just eliminates the VisGroup, making them individual unlinked objects.

You can also double-click the name and a hand will appear next to the VisGroup. This is how you set which VisGroups are visible and which are not.

To add new objects to an existing VisGroup, select the group in the **Filter Control**, then select objects in 2D or 3D views, and then press the **Assign** button. New objects will be added to the selected VisGroup. You can also drag one group onto another group: this will cause the first group to be merged with the second group, and the resulting group will keep the name of the second.

If you make the furniture group selected, then select it if the **Filter Control** and press the **Mark** button; the group will be highlighted in the 2D and 3D views.

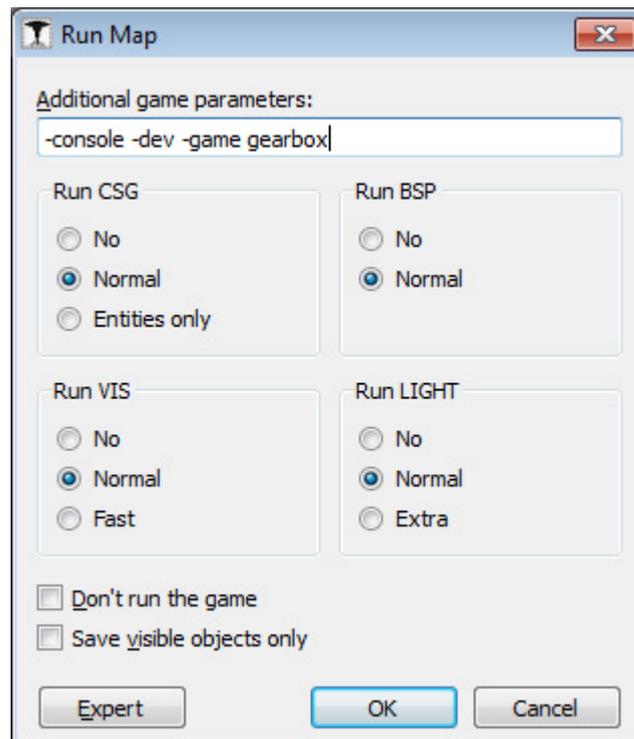
Pressing the **Purge** button removes any VisGroups from the Filter Control dialog that do not actually contain any objects (which is possible if you have deleted objects or made one VisGroup part of another).

Note that in J.A.C.K., multiple VisGroups per object are allowed, that can sometimes mess up the visibility determination of VisGroups in the Filter Control. To remove the VisGrouping from an object, open its properties (**Alt+Enter** or double-click), switch to the **VisGroup** tab, and then click on “(no group)” checkbox.

3.9. Compiling and Running Your Level

If you are going to use J.A.C.K. to compile your maps, there are two different compile modes for you to use: the normal compile mode and the advanced compile mode (Expert Mode).

The normal compile mode allows for one button compiling with no further setup other than what has been mentioned in the [Setup Guide](#).



Additional game parameters: this is where you specify the additional parameters for the game executable. If you are compiling a map for a mod, be sure to include the `-game <moddir>` parameter as shown in the above picture. `-dev` and `-console`, also shown, are good parameters to use when testing maps (see below).

Run CSG, BSP, VIS, and LIGHT: these settings allow you to define which tools will be run along with a couple of their most basic parameters.

Don't run the game: if this option is checked, the game will not be started after the map has finished compiling.

Save visible objects only: if this option is checked, only objects that are not hidden will be compiled into the map.

Press the **Expert** button to go into [Advanced Compile Mode](#), which gives you much more control on how your level will be pre-processed, compiled and distributed. And some useful game parameters:

- **-dev** — this puts the game in developer mode, providing extra debugging information for you when you're testing your map.
- **-console** — this enables the in-game console which is accessible by pressing the tilde (~) key.
- **+set deathmatch 1** — this starts the game in multiplayer (deathmatch) mode. Some games and mods, like Counter-Strike, require this.
- **-game <moddir>** — this should be used when you're making a map for a mod. For example, if you're making an Opposing Force map, you'd need to use the “-game gearbox” parameter.

4. Advanced Topics

4.1. Working with Models

Many games use triangle models to make environment more detailed. J.A.C.K. offers some features to make working with models much more comfortable.

Placing models is very simple: select a corresponding entity class (e.g. `cycler`, `env_model`, `misc_model` or whatever; refer to the game's documentation which entities are good for that) and place the entity into the map. Models are always point entities; see [Point Entity Placement](#) for more information on how to create such entities.

Now you should open [Object Properties](#) (**Alt+Enter**) and find a key that actually assigns a model to the entity (it also depends on the game). Press the ... (three dots) button nearby to browse for a model file. J.A.C.K. will load custom models on demand, and keep them in its cache memory until you create or open a new project; in that case the memory may be cleared and models are not guaranteed to be resident.

Some games (e.g. Half-Life) define specific attributes of model appearance: body, skin, etc. J.A.C.K. does his best to display such attributes. This means that, unlike Valve Hammer Editor/Worldcraft, you can preview body groups and skin families directly in the editor.

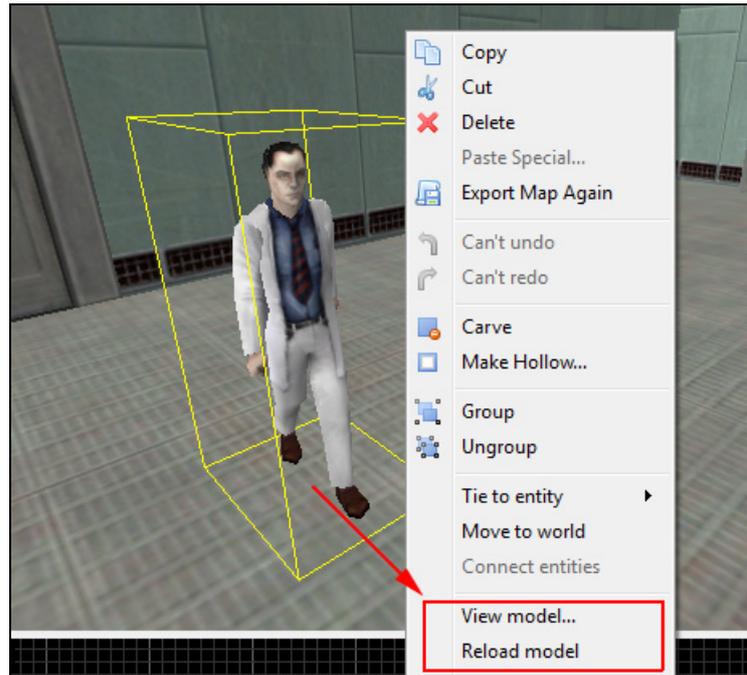
Some models can animate in 3D views, and such behavior can be disabled in the [Configuration Dialog](#). Even if the entity doesn't define a specific animation sequence key (usually "sequence"), you can enter it manually after turning off Smart Mode in the **Object Properties**. Often this doesn't affect models in game, but useful for preview purposes.

Although models' textures are loaded by J.A.C.K., and even though sometimes they are external files and effects (in Quake III), the editor will hide them. Such textures won't be accessible in the **Texture Browser** unless added explicitly to the game configuration.

If you have the **Alt+drag to scale models** option enabled, you can scale a model in the same way you scale solids with the Selection Tool. However, models are scaled uniformly in all directions (no per-axis scale factors). Vertex Manipulation for models is also not available. Note that some games don't support scaled models, it actually depends on the engine and certain entities. E.g. original Half-Life doesn't support scale on cyclers; but Spirit of Half-Life mod introduces a new entity, `env_model`, to be used instead of cyclers, and it does support scaling. Quake III supports scaled `misc_model`s with `q3map2` map compiler.

You may preview wireframe model meshes in 2D views; to turn this off, use the **Draw Models** option in [3D Views](#) configuration page, or the **Show/hide models in 2D** button on the [Map Render](#) toolbar.

Entities with models have some additional useful commands in the context menu:

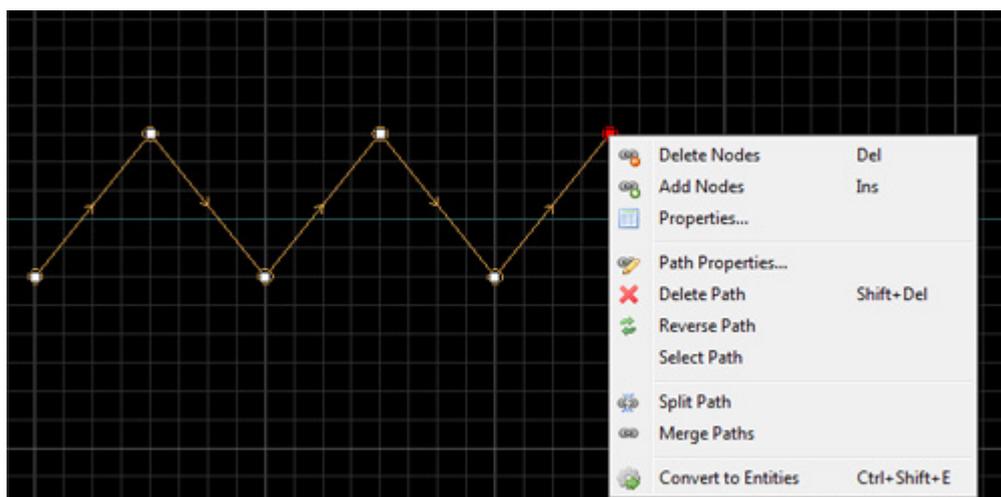


View model...: opens the model currently selected in model viewer (you have to define a path to your favorite model viewer for a certain game in [Game Configurations](#), because it is not set automatically; otherwise this command will be absent).

Reload model: simply reloads currently selected model from disk; use this command if you've modified the model elsewhere, and want to see the changes without restarting J.A.C.K.

4.2. Path Tool

The Path Tool greatly simplifies the creation of paths for trains and monsters. It allows you to create dynamic paths that you can add and delete nodes to.



With the Path Tool selected:

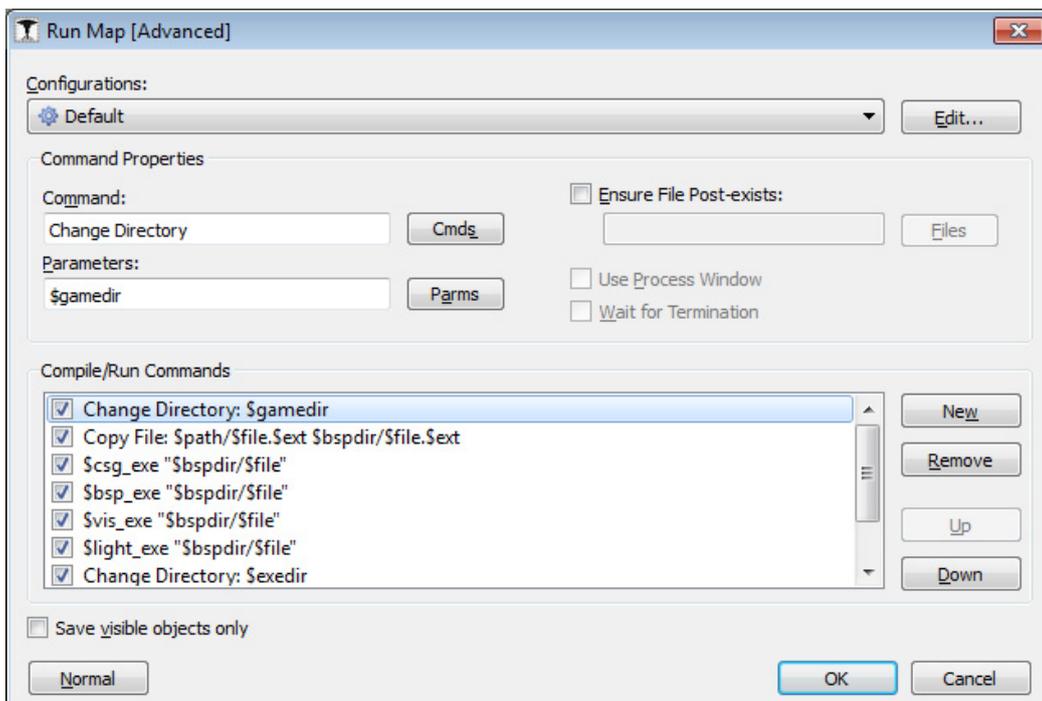
1. Hold **Shift** and the left mouse button, and click in a 2D view. This creates the first path node.

2. A dialog box will pop up where you can set the name of the path, the path direction (one way, circular, or ping-pong), and the entity type (typically “path_corner”).
3. To add new nodes to the end of the path, click on one of the existing path nodes, hold **Shift** and click the left mouse button. Do this as needed until you have your path complete.
4. To insert new nodes to the path, click on the existing path node and press **Ins**. The new node will appear before the selected node.
5. To delete nodes, select them and press **Del**. The path will be relinked automatically. To delete the whole path, click on one of the existing path nodes, and press **Shift+Del**.

You can also convert path to a connected set of entities (**Ctrl+Shift+E**) to edit particular nodes as entities. Please note that you cannot convert a set of entities back to a path!

4.3. Compiling in Expert Mode

The advanced compile mode (“Expert Mode”) gives you much more freedom over how the compile process is performed.



TODO: a relatively large tutorial will be placed here, once I have some time to write it.

For now, the only option you have is to look at the screenshot of the **Run Map [Advanced]** dialog and hope that everything is self-explanatory (well, I know it is not).

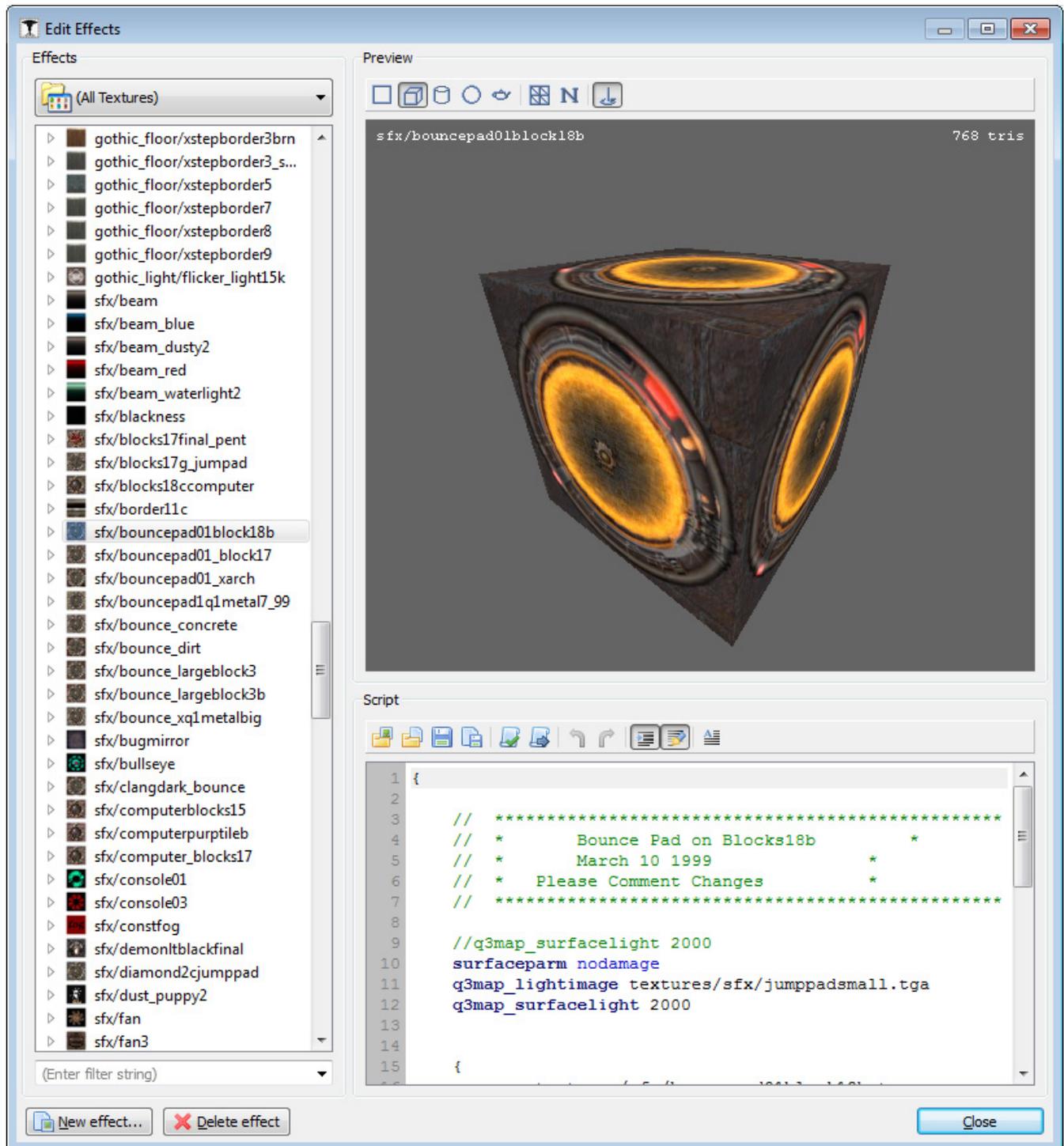
4.4. Creating Curves (Quake III)

TODO: a relatively large tutorial will be placed here, once I have some time to write it.

4.5. Effects Editor (Quake III)

TODO: a relatively large tutorial will be placed here, once I have some time to write it.

For now, the only option you have is to look at the screenshot of the **Effects Editor** dialog and hope that everything is self-explanatory (well, I know it is not).



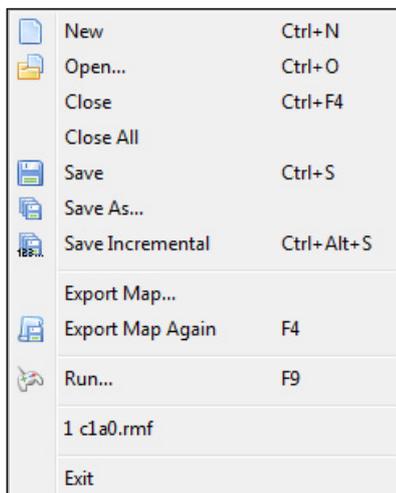
For a complete description of Quake III Arena effect script format, refer to:

http://q3map2.robotrenegade.com/docs/shader_manual/

5. Reference

5.1. Menus

File



The File menu is mostly self explanatory: it deals with all of J.A.C.K. file operations. The only commands that may require a bit of explanation are:

Save Incremental

This command is very similar to the Save command, but it automatically adds version number to the output file name. Such behavior is familiar to 3DSMax users; it enables easy creation of checkpoints during prolonged project development.

Export Map...

This is used to export your map into a MAP file (“.map”).

J.A.C.K.'s normal map format is JMF, but the compile tools will only read MAP files. The JMF file contains information like camera positions, VisGroups, and grouping information, whereas the MAP file does not. This will be done automatically when you use J.A.C.K. to compile your map, but if you use a batch file or external compile program to compile, you'll need to use this function.

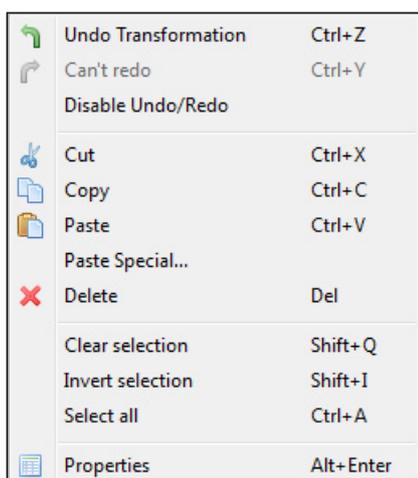
Export Map Again

Once you've used Export Map once, you can automatically export your map to the same location and name by selecting this function or by using its hotkey F4.

Run...

This is used to open a compile dialog; after the compile process you will be able to run the game on this level.

Edit

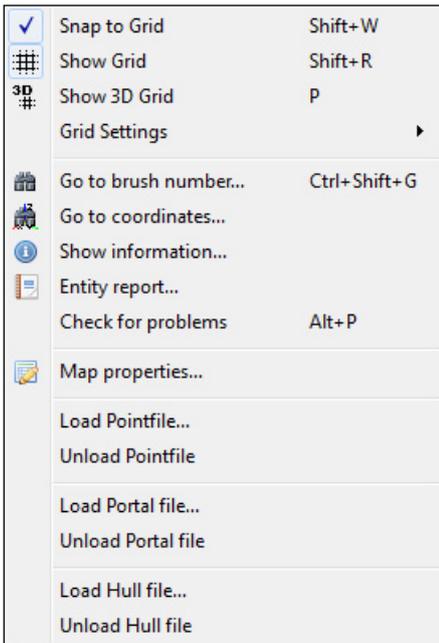


Everything here is pretty self explanatory. The Properties function (which is mostly handily used through its **Alt+Enter** hotkey) is used to view the properties of world and entity objects.

The **Paste Special...** command brings the [Paste Special](#) dialog which draws some interesting options for a level designer.

Disable Undo/Redo can be used to flush the contents of the undo buffer to free some memory. To achieve this, click it, and then click **Enable Undo/Redo** this item will be transformed into.

Map



The Map menu contains a number of functions related to the display of the map, as well as global viewing and manipulation of map contents and properties.

Grid Options

Both **Snap to Grid** and **Show Grid** are fairly self explanatory. **Show 3D Grid** toggles grid overlay on brushes in 3D views. **Grid Settings** allow you to adjust the size of the grid (although using the [and] hotkeys let you do this much quicker.

Go to brush number...

This brings up the [Go to Brush](#) dialog which allows you to go directly to a brush or entity by its id number. This is handy if you're using compile tools that will report a brush/entity number

along with an error. You have the option of searching through all brushes or visible brushes only; this should be set according to how the map was compiled.

Go to coordinates...

This brings up the [Go to Coordinates](#) dialog which allows you to go to coordinates specified in “X Y Z” format.

Show information...

This brings up the [Map Information](#) dialog.

Entity report...

This brings up the [Entity Report](#) dialog.

Check for problems

This brings up the [Check for Problems](#) dialog.

Map properties...

This brings up an [Object Properties](#) dialog with the “Worldspawn” properties. The Worldspawn is an entity that defines the characteristics of the world. This is the only way to them.

Load and Unload Pointfile

When the compile tools report a leak in your level, you can use these functions to load and unload the leak file (“.pts” or “.lin”) directly into J.A.C.K. 2D and 3D views. A colored gradient line will eventually pass through the leak, and can be traced easily in the editor.

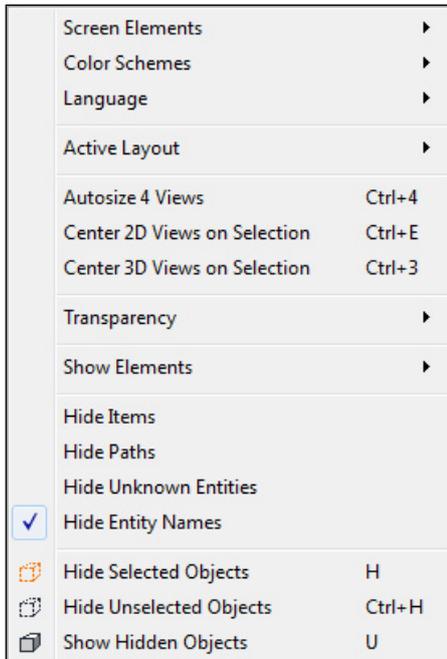
Load and Unload Portal file

These functions can be used to load and unload the portal file (“.prt”) to examine portals generated by BSP compiler. This can be useful to access BSP complexity and find possible VIS flow problems.

Load and Unload Hull file

These functions can be used to load and unload the hull file (“p0”, “.p1”, “.p2”, “.p3”) generated by CSG compiler; thus this is Half-Life specific feature. This may be helpful in searching for clipping hull problems.

View



The purpose of the functions in this menu is to control what does and doesn't appear in J.A.C.K.'s 3D and 2D views. It also controls the interface of J.A.C.K., e.g. colors and language of the UI. For more ways to customize the appearance of J.A.C.K., refer to the sections on the [Configure J.A.C.K.](#) dialog options.

Screen Elements

This submenu allows you to show and hide J.A.C.K.'s [toolbars](#).

Color Schemes

This submenu contains a list of predefined UI color schemes, along with a command to define a custom UI color scheme.

Language

This submenu contains a list of available UI languages. You can change the localization of the editor on the fly; restart is not needed.

Active Layout

Selecting any of these will turn the currently selected view into whatever you've selected: X/Y (top), Y/Z (front), or X/Z (side) view. Selecting any of the rest will turn the currently selected view into wireframe, filled polygons (flat shaded colored polygons), textured polygons (textured as they would be in the game), or shaded textured polygons (same as textured, but shading is also applied for better feeling of depth complexity).

Autosize 4 Views

This function only works when you *aren't* using independent windows. The four views will be set to equal sizes based on the current available screen space.

Center 2D/3D Views on Selection

Choosing this function will center all of the 2D/3D views on whatever you currently have selected.

Transparency

This submenu allows toggling transparency of special textures and/or entities.

Show Elements

This submenu allow toggling display of targetname-target entity connections, wireframe models in 2D views, realistic sky, texture/effect animations, faces with special textures (nodraw/clip/etc.) and UV grid (texture image borders in 3D views).

Hide Items and Hide Paths

These functions are used to hide items (entities with an “Item” flag in FGD; or with item_XXX classname) and paths (entities with an “Path” flag in FGD; or with path_XXX classname).

Hide Unknown Entities

This function is used to hide entities not described in FGD files of the current game configuration.

Hide Entity Names

This function toggles rendition of text labels displaying “classname” and “targetname” fields of entities.

Hide Selected/Unselected Objects

This function is to hide the currently selected/unselected objects.

Show Hidden Objects

This function makes all hidden objects visible.

Tools

	Carve	Ctrl+Shift+C
	Merge	Ctrl+Shift+M
	Make Hollow...	Ctrl+Shift+H
	Group	Ctrl+G
	Ungroup	Ctrl+U
	Make Detail	Ctrl+Shift+D
	Make Structural	Ctrl+Shift+S
	Tie to entity...	Ctrl+T
	Move to world	Ctrl+W
	Connect entities	Ctrl+K
	Texture application...	
	Replace textures...	
	Edit effects...	F3
	Texture lock (move+rotate)	Shift+L
	Texture lock (scale)	Ctrl+Shift+L
	Texture lock (UV coordinates)	
	Align to view	Ctrl+Shift+A
	Snap selected to grid	Ctrl+B
	Snap selected to grid individually	Ctrl+Shift+B
	Transform...	Ctrl+M
	Align objects	▶
	Flip objects	▶
	Rotate objects	▶
	Synchronize with SteamCloud	F5
	Automatic configuration...	Ctrl+F2
	Options...	F2

Carve

Use the Carve tool to subtract the shape of the selected object from the surrounding area. For more information, see the [Reshaping Solids](#) chapter.

Merge

Merge two or more solids that form a convex hull, into a single solid. For more information, see the [Reshaping Solids](#) chapter.

Make Hollow...

Make Hollow turns a solid object into a hollow shell of the same shape. You must have a brush selected for this to work. You will be asked to supply a value for wall thickness. Positive numbers keep the hollowed brush the same size, and negative numbers will cause it to expand. See the [Hollow](#) dialog description for the details.

Group/Ungroup

Group/Ungroup binds two or more objects together so they may be acted upon simultaneously. This is not the

same as `VisGrouping`. The two functions differ in that `grouping` provides a way of physically binding a group of objects together, while `VisGrouping` enables you to organize objects into a group that may still be worked upon as separate objects. For more information, see [Grouping and VisGrouping](#).

Make Detail/Structural

These commands allow you alter the contents of selected solids, effectively adding or removing the “Detail” flag. These options are visible only in game profiles that support detail brushes (e.g. Quake III). In other games these functions may be not applicable; e.g. in Half-Life detail brushes are marked by turning them into a pseudo-entity `func_detail`.

Tie to entity/Move to world

These two commands allow you to designate an object as part of the world (a structural piece, like a wall), or as an entity (a door, for example). When you tie a brush to an entity, a dialog box pops up, allowing you to choose which entity, and to modify the entity properties. This dialog will contain brush based entities only (doors, plats, triggers, etc).

Connect entities

The command connects two entities with a `targetname-target` link. See the [Names and Targets](#) chapter for more information.

Texture application...

This switches the current tool to the **Face Properties** dialog which allows you to modify the properties of individual brush faces. See [Texturing Solids](#) for more information.

Replace textures...

`Replace Textures` brings up the [Replace Textures](#) dialog box which lets you search for and either mark or replace all textures of a certain type.

Edit effects...

This command opens the **Effect Editor**. Not all the games support effects; currently only Quake III and Volatile Engine support them. See [Effects Editor](#) for the detailed description.

Texture lock (move+rotate)

This option toggles texture lock mode on and off. Texture lock allows you to move or rotate a brush without disturbing its texture alignment.

Texture lock (scale)

This is an additional texture lock which retains texture coordinates during brush scaling.

Texture lock (UV coordinates)

This option toggles a special texture lock mode, also known as “UV lock”. When this mode is on, J.A.C.K. attempts to preserve texture coordinates at vertices, modifying texture axes where appropriate. This can result in skewed texture alignment, which can be very useful on walls modified in “shear”

mode. Note that this lock should be turned on only during such operations like shear; otherwise it can severely damage texture alignment.

Align to view

This is a special texture alignment mode. It looks like projecting a texture from the camera face, as if you were a slide projector and the texture is a slide. It's useful for painting a texture onto a cliff or other irregular things. Multiple solids or faces may be selected before using this function.

Snap selected to grid

Not all objects will align perfectly to the grid. Some objects, such as entities, are occasionally difficult to align with the grid. This command will snap the corner of the bounding box of selected solids and/or entities to the grid. Subsequent snap commands will cycle through different corners of the bounding box.

Snap selected to grid individually

Same as above, but snaps by individual bounding boxes instead of the bounding box of the whole selection. In addition, it snaps point based entities by their origin (center point) instead of a corner of the bounding box.

Transform

This brings up the [Transform](#) dialog which allows you to rotate, scale, and move an object a precise value in any of the X, Y, or Z planes.

Align objects

Align objects lets you align a set of objects to a specified side of the bounding box of the whole selected set.

Flip objects

Flip objects allows you to flip objects horizontally or vertically (or both, to create a mirror image of the object). Horizontal and vertical are defined relative to the last active 2D view.

Rotate objects

Rotate objects lets you to rotate the selection either clockwise or counter-clockwise. The clock direction is defined relative to the last active 2D view.

Synchronize with SteamCloud

This will bring up the [Synchronization with SteamCloud](#) dialog (Steam version only).

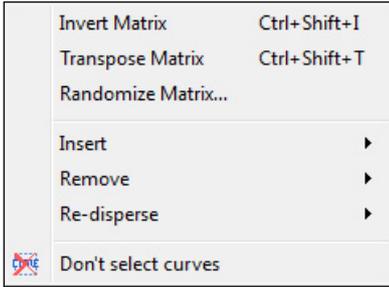
Automatic configuration...

This will bring up the [Automatic Configuration](#) dialog (Steam version only).

Options...

This will bring up the [Configure J.A.C.K.](#) dialog which allows you to modify a number of J.A.C.K.'s behaviors.

Curve



This menu is available only for game profiles which support [curved surfaces](#) (Quake III, Volatile).

Invert Matrix

This command inverts matrices of the curves selected, effectively flipping orientations of their faces.

Transpose Matrix

This command transposes matrices of the curves selected, converting rows to columns, and vice versa. It does not affect visual appearance though; useful in conjunction with other tools.

Randomize Matrix

This brings the dialog which allows randomizing of curve points; useful for curve-based terrains.

Insert

This submenu contains commands to prepend or append rows or columns to the selected curves.

Remove

This submenu contains commands to remove first or last rows or columns from the selected curves.

Re-disperse

This submenu contains commands to re-disperse (uniformly redistribute) rows or columns in the selected curves.

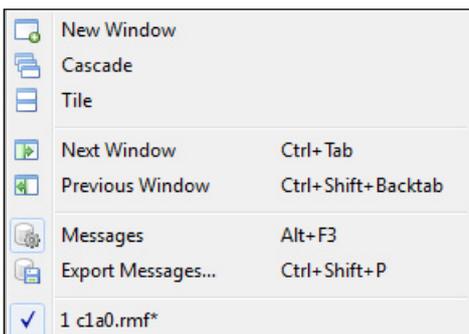
Don't select curves

When this option is toggled, curves can't be selected in 2D and 3D views.

Plugins

If your plugins define any commands, they will be listed in the **Plugins** menu. There are some useful plugins shipped with J.A.C.K., e.g. **Round Coordinates** (to ensure Valve Hammer Editor/Worldcraft map export behavior, which rounds coordinates to integers during the map export) and **Extract Textures** (especially useful for Quake, where all the textures reside inside compiled BSP files, and no WAD files are provided within the game).

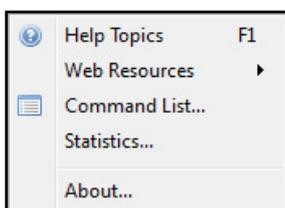
Window



Everything here is self explanatory. The **Messages** function will toggle visibility of a panel with any errors, warnings and information messages spit during J.A.C.K.'s workflow. You can save these messages to a text file using an **Export Messages...** command.

The **New Window** function is useful for creating new windows when you're using independent window configurations.

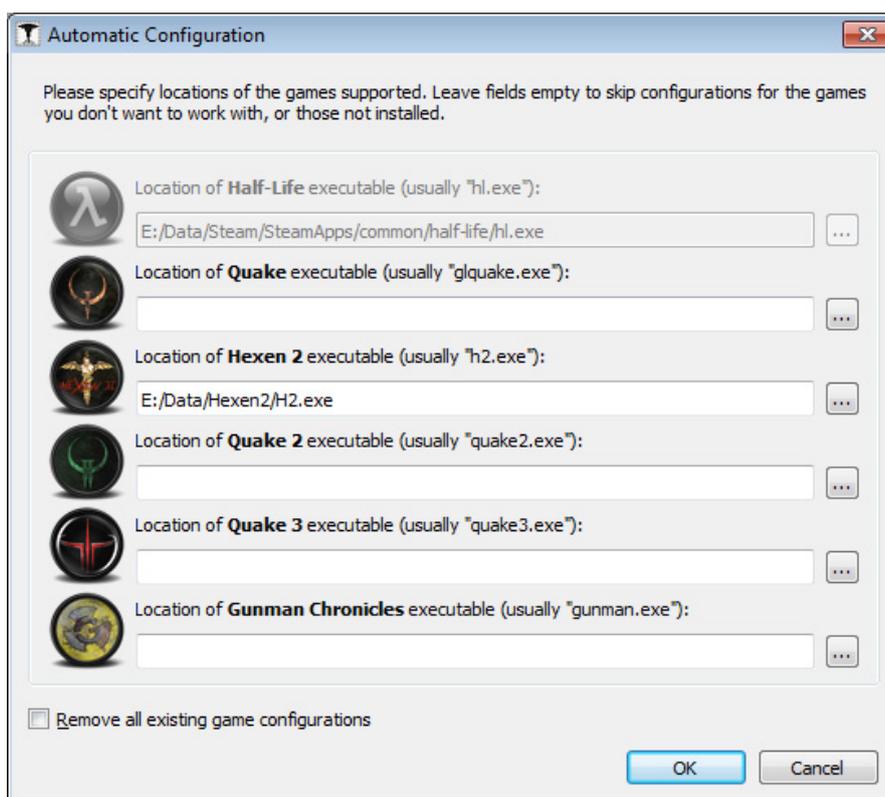
Help



This menu provides easy access to a number of help sources, including several websites that will provide J.A.C.K. editing information. The **Help Topics** command (invoked by **F1** hotkey) opens this manual. The **Command List...** function displays a list of hotkeys; the same list exists in this manual (see [Hotkey Reference](#) chapter), but is more handy since doesn't require opening the whole help file. The **Statistics...** command opens the dialog with *Steam Stats* (this command is available in the Steam version of J.A.C.K. only).

5.2. Dialogs

Automatic Configuration



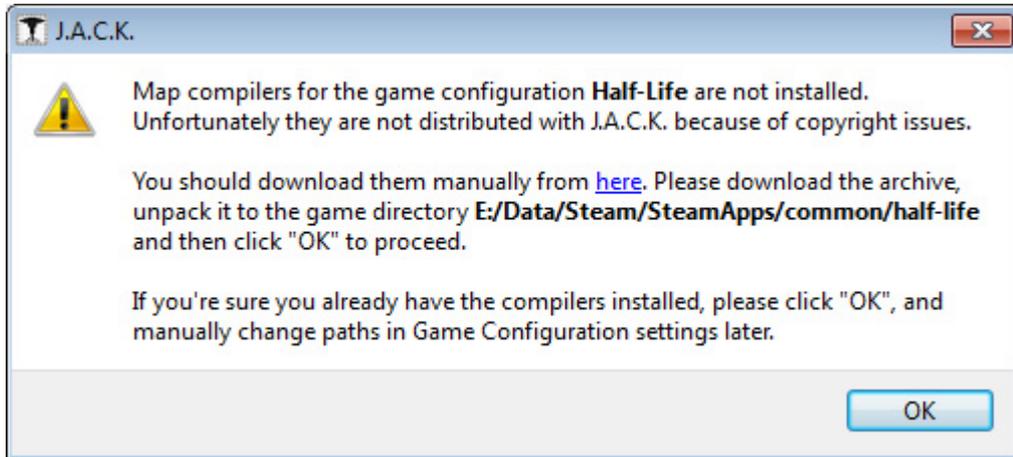
Hotkey: **Ctrl+F2**

Note: this dialog exists in the Steam version of J.A.C.K. only!

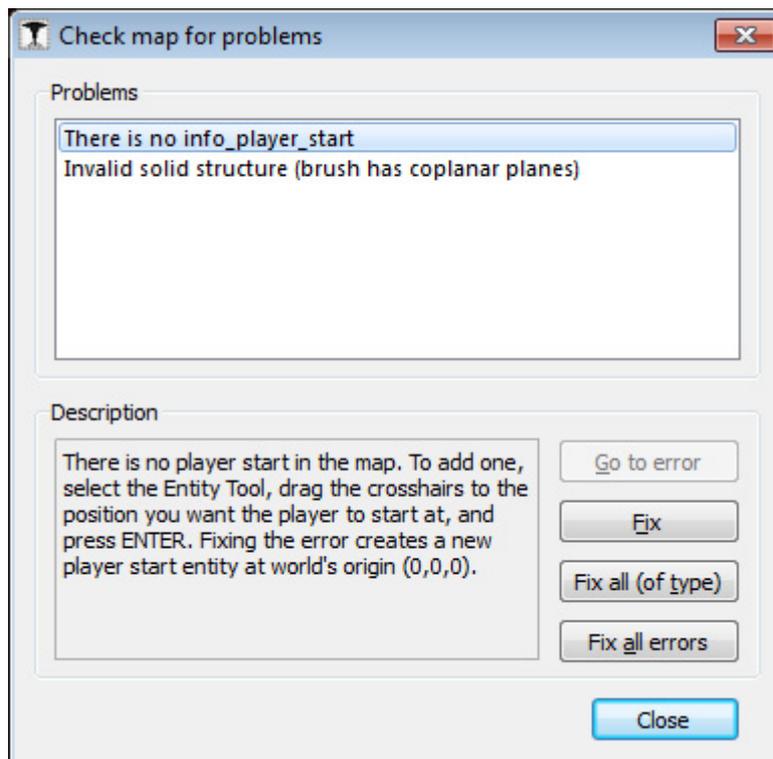
This dialog can be used to automatically create Game Configurations for the supported games installed. You can specify paths to their executables and press **OK** to create configurations for them. J.A.C.K. tries to detect paths automatically. If the game is grayed, this means that the game configuration for this game already exists.

Remove all existing game configurations: perform “clean setup”, i.e. first remove all the existing game configurations, and then create new ones. Caution, this operation cannot be undone!

Sometimes J.A.C.K. cannot complete the game configuration, because some tools are not distributed with it for some reason. In this case, you'll get instructions how to complete the configuration process. E.g. this message explains how to install map compilers for Half-Life, and provides a user with the download link for the current operating system and the version of the editor.



Check for Problems



Hotkey: **Alt+P**

Opening this dialog will cause J.A.C.K. to run through your map and notify you of any errors it finds. When you select one of the errors, the entity/solid will be highlighted (if applicable) and three options become available to you. **Go to error** will center the 2D and 3D windows on the selected error

object, **Fix** will attempt to fix that error, and **Fix all (of type)** will fix all errors of the selected type. In addition, there is the button **Fix all errors**, which attempts to fix all the errors found.

Here is a list of some common errors that may occur in your maps:

- **There is no info_payer_start**

There is no player start in the map. To add one, select the Entity Tool, drag the crosshairs to the position you want the player to start at, and press Enter. Fixing the error creates a new player start entity at world's origin (0,0,0).

- **Unmatched 'target' field**

This entity has a 'target' value that isn't matched by a 'targetname' value in another entity. To fix the problem, go to the entity, open [Object Properties](#) (**Alt+Enter**) and select a proper target name from a drop-down combo.

- **Entity contains duplicate keyvalues**

The entity contains duplicate keyvalues. Fixing the error removes the duplicates.

- **Entity contains double quotes in keyvalues**

The entity contains double quotes (") in its keyvalues. This will probably cause errors when parsing text format such as *.map, by the editor or compile tools. Fixing the error replaces forbidden double quotes with single quotes (').

- **Bad 'targetname' field**

This entity has a 'targetname' value that contains starting character reserved as "use prefix". Fixing the error removes the characters that could be referred to as such prefixes.

- **Invalid texture**

The solid is using an invalid (missing) texture. This error usually occurs when some texture packages used on the map were not added to the current game configuration. Fixing the error replaces the texture with currently selected texture (but only with a valid one).

- **Invalid texture**

The solid is using an invalid (missing) texture. This error usually occurs when some texture packages used on the map were not added to the current game configuration. Fixing the error replaces the texture with currently selected texture (but only with a valid one).

- **Texture axis perpendicular to face**

The face has a texture axis that is perpendicular to the face, probably caused by rotating an object with texture locking disabled. This could also indicate an invalid face caused by vertex manipulation.

- **Solid entity is empty**

This solid entity contains no solids. Fixing the error deletes the entity.

- **Solid with no faces**

This solid contains no faces. Fixing the error deletes the solid.

- **Invalid solid structure**

The solid has invalid structure, probably as a result of vertex manipulation. The exact problem with the solid is shown in brackets, e.g. “brush has coplanar faces”. You will need to either fix it, or if this is not possible, rebuild it.

- **Invalid solid contents**

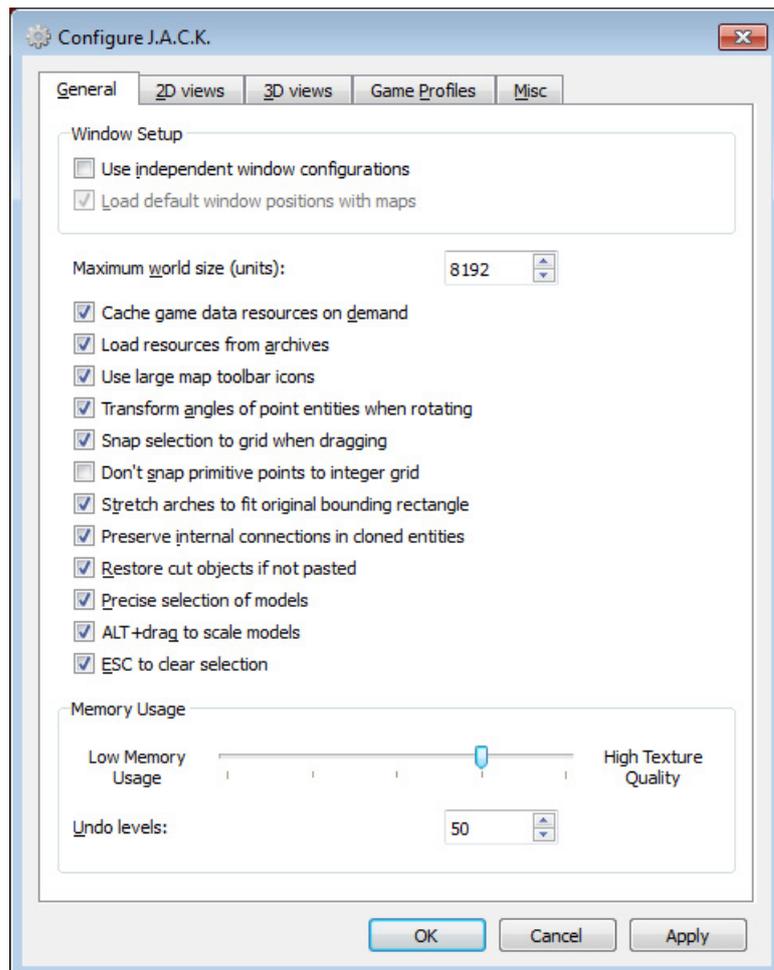
The faces in the solid have different content types, which may result in 'Mixed face contents' compiler error. Fixing the error assigns the contents of the first face to all faces of the solid.

Configure J.A.C.K.

The configuration dialog (**F2**) contains multiple tabs and enables J.A.C.K. customization in many aspects, along with creation of [Game Configurations](#).

A. General Options

This tab let you configure the most common J.A.C.K. Options.



- **Use independent window configurations:** J.A.C.K.'s Independent Window Configuration is simple to use and extremely handy. It lets you customize your windows setup virtually any way

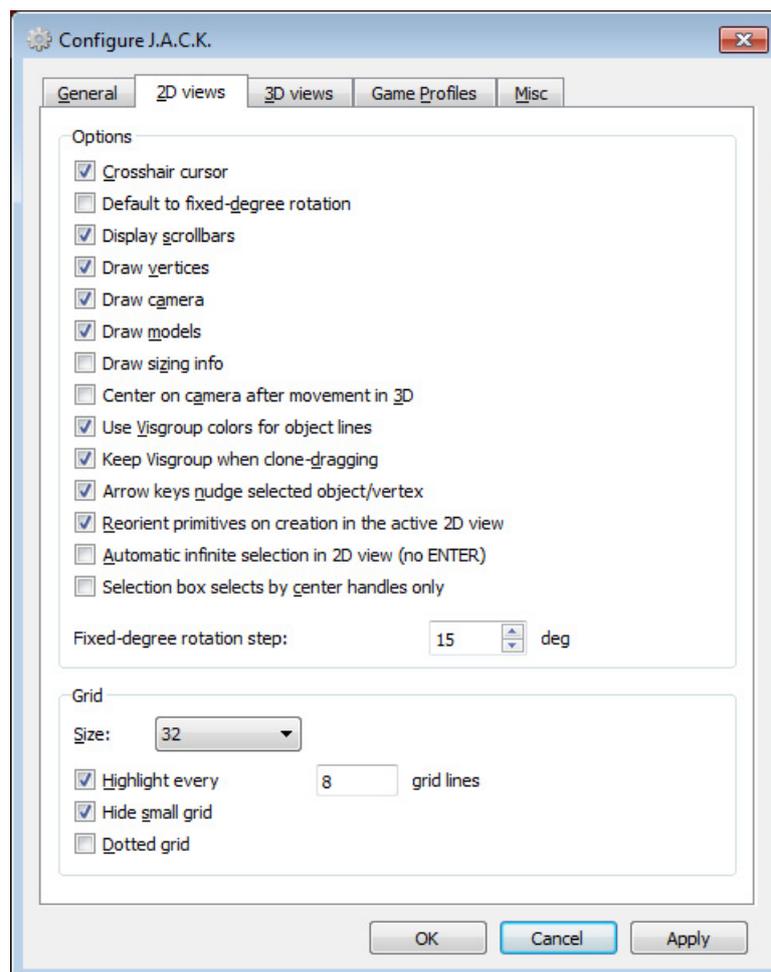
you'd like. To add new windows, select New Window from the Window menu. Windows can be resized any way you like.

- **Load default window positions with maps:** if this is checked, the last used window setup will be loaded when you load a map. If this is not checked, you can still load the window setup manually using the **Load Window State** button on the [Map View](#) toolbar.
- **Maximum world size (units):** maximum size of available space in 2D views for level construction; note that this is also a compiler limit. E.g. default Half-Life maximum world size is 8192 units.
- **Cache game data resources on demand:** load textures/models/etc when you create a map for a certain game configuration for the first time. If this option is unchecked, J.A.C.K. will precache all the game configurations on startup. This can be very slow if you use more than one game configuration; therefore it is recommended to leave this option checked.
- **Load resources from archives:** load models and sprites from game archives (.pak, .pk3) if they are not unpacked.
- **Use large map toolbar icons:** uncheck this option to use small icons on [Map Tools](#) toolbar, like any other toolbar uses.
- **Transform angles of point entities when rotating:** change point entity angles to match the rotation angle; useful to rotate platforms with monsters placed on them, etc.
- **Snap selection to grid when dragging:** snap to current grid size on dragging; uncheck to snap to the minimum (one pixel) grid instead.
- **Don't snap primitive points to integer grid:** don't snap vertices for newly created primitives.
- **Stretch arches to fit original bounding rectangle:** when this is set, arches will fill the entire bounding box that you've defined; otherwise they will treat the bounding rectangle as a "full circle" area and only take up a portion of that area.
- **Preserve internal connections in cloned entities:** targetname-target links will be maintained unique during copy-paste operations (also see [Paste Special](#)).
- **Restore cut objects if not pasted:** cut objects that were not pasted subsequently, will reappear.
- **Precise selection of models:** test again the model's mesh precisely when selecting a model in 2D or 3D views; if unchecked, use bounding box instead.
- **ALT+drag to scale models:** enable mouse dragging while holding Alt to scale models; note that not all the game profiles support scale for models; e.g. original Half-Life does not, but Quake III with q3map2 compiler does.
- **ESC to clear selection:** enables Q3Radiant-style behavior, where pressing **Esc** clears the current selection.

- **Memory usage:** you can balance between low memory usage and high texture quality. It is recommended to place the slider at the 4th position from the left; this is the most balanced setting.
- **Undo levels:** the undo levels default to the last 50 actions. This can take up quite a bit of memory. Setting this to a lower value may improve performance slightly.

B. 2D Views

The 2D Views options allow you to configure a number of J.A.C.K.'s behaviors. Settings here will depend largely on personal preference.



Options

- **Crosshair cursor:** when this is enabled, the cursor will turn into a crosshair when it is over a brush.
- **Default to fixed-degree degree rotation:** when rotating a brush, the rotations will be in fixed degree increments rather than 1 degree. The **fixed-degree rotation step** can be customized below, and defaults to 15 degrees.

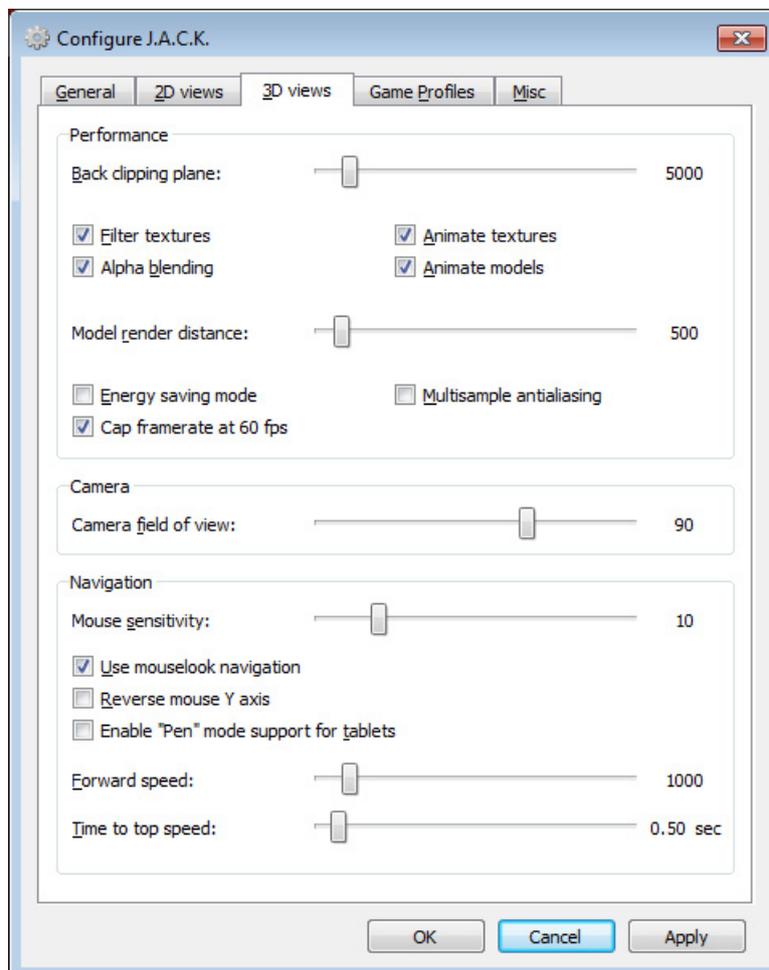
- **Display scrollbars:** if this is enabled, each 2D view will have a set of scrollbars to allow you to move the view around. If this is not enabled, you can still move around the 2D views by holding down the **Space bar** and clicking and dragging inside a 2D view with the mouse.
- **Draw vertices:** when enabled, brush vertices will be drawn in the 2D views.
- **Draw camera:** when enabled, a blue eye representing the current 3D view camera position, will be drawn in the 2D views; don't mix it with [user-defined cameras](#) of the **Camera Tool**.
- **Draw models:** when enabled, model meshes will be drawn in 2D views, instead of simple bounding boxes.
- **Draw sizing info:** when enabled, helpful sizing information will be drawn near the selected objects.
- **Center on camera after movement in 3D:** use this option if while using the mouse to move around the 3D view while in camera mode, you want the 2D views to automatically center on the new camera position.
- **Use VisGroup colors for object lines:** when this is enabled, the object lines for any non-entity brush will be the color specified in its [VisGroup](#) when viewed in the 2D views. Objects not associated with a visgroup will still appear in generic brush/entity color.
- **Keep VisGroup when clone-dragging:** select this feature if you want newly cloned objects to retain grouping properties of the original object when clone dragging (selecting an object, holding down **Shift**, and dragging the object).
- **Arrow keys nudge selected object/vertex:** when this is enabled, you can move a selected object or vertex by using the arrow keys. The movement is performed relative to the current active 2D view.
- **Reorient primitives on creation in the active 2D view:** when this is enabled, primitives will be oriented with their "top" shown in whichever 2D view they were created in. This is most easily shown when creating a cylinder. With this option disabled, the cylinder will be oriented so that the round part shows up in the XY 2D view, regardless of which view you created it in. This is mostly just a time saving feature.
- **Automatic infinite selection in 2D windows (no ENTER):** when this is enabled, dragging a selection box over an object or objects will immediately cause them to be selected (normally you would need to press **Enter**). This option can be turned on and off from the [Map Operations](#) toolbar as well.
- **Selection box selects by center handles only:** when using a selection box to select a number of objects, it defaults to selecting any object the box touches. With this option enabled, it will only select those objects whose "handles" (the small "x" at the center of each object) are within the selection box. This option can also be turned on and off from the [Map Operations](#) toolbar as well.

Grid

- **Size:** this determines the initial grid size when you start a new map. The options are power-of-2 values, from 1 to 512 units. You can change the grid size while editing by pressing "[" and "]".
- **Highlight every 8 grid lines:** this will highlight one line every 8 grid lines. The actual number can be set to whatever you like.
- **Hide small grid:** if the grid size is smaller than 4 pixels, it will generally appear as a gray background instead of a grid, unless zoomed in extremely close. When enabled, this option turns off the grid when it gets that small.
- **Dotted grid:** enabling this option causes the grid to be displayed as a series of dots, rather than lines. This may help if you find that the grid lines blend into your map lines.

C. 3D Views

The 3D Views options allow you to set the behavior of J.A.C.K.'s 3D views.



Performance

- **Back clipping plane:** this sets the distance in units at which J.A.C.K. will stop drawing in its 3D view. If you are finding the 3D performance to be too slow, set this to a lower value.

- **Filter textures:** when this feature is enabled, bilinear filtering is done on textures resulting in a smoother non-pixelated look when viewed close-up. Disabling this may result in a slight speed increase, or a dramatic speed increase on older video cards. You can also press **F** in a 3D view to toggle this option.



- **Alpha blending:** enable transparency for map objects rendered that require it.
- **Animate textures:** enable texture and effect animations.
- **Animate models:** enable model animations (for model formats that support animations).
- **Model render distance:** this sets the distance in units at which J.A.C.K. will replace models with bounding boxes. If you use many models and are finding the 3D performance to be too slow, set this to a lower value.
- **Energy saving mode:** lowers energy consumption by J.A.C.K.; useful if you are editing levels on a laptop. This may also lead to much less smooth camera movement and animations.
- **Cap framerate at 60 fps:** set 60 fps (frames per second) limit in 3D views to keep CPU and GPU payload at low values, but unlike the previous option, this shouldn't much affect smoothness.
- **Multisample antialiasing:** apply full-scene multisample antialiasing (MSAA) to the 3D view; useful if you're going to take some screenshots. Not recommended for a general workflow, since it may affect accuracy of selection in 3D views.



Camera

- **Camera field of view:** field-of-view (fov) angle of the 3D camera, in degrees; values between 80 and 120 are recommended. Default FOV in J.A.C.K. along with many games is 90 degrees.

Navigation

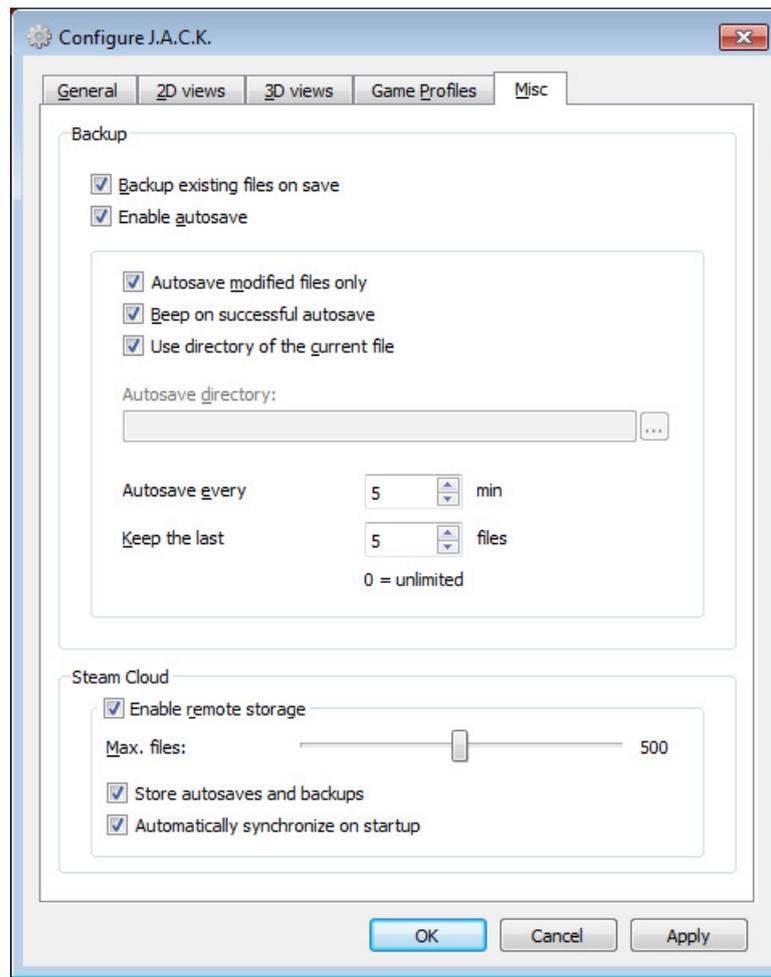
- **Mouse sensitivity:** defines how sensitive is mouse in 3D navigation, just like in games.
- **Use mouselook navigation:** when this is enabled, "mouselook navigation" will be available. The **Z** hotkey toggles this mode on and off. While on, moving the mouse will change the viewing angle in the 3D view, and the WASD keys can be used for forward, left, backward, and right movement.
- **Reverse mouse Y axis:** when you are using the mouse to navigate through the 3D view, if this is enabled pulling the mouse backwards will cause the view to look upward, and moving the mouse forward will cause the view to look downward.
- **Forward speed:** you can set the maximum forward movement speed here.
- **Time to top speed:** this is the amount of time it takes to go from 0 to the above maximum movement speed (basically this sets the acceleration).

D. Game Profiles

This tab let you set up new game configurations or edit existing. It is quite well described in the [Manual Setup](#) chapter.

E. Misc

This tab let you configure miscellaneous J.A.C.K. Options.



Backup

- **Backup existing files on save:** make backups on every save, replacing the third letter in file suffix with x (e.g. jmf → jmx, rmf → rmx, map → max).
- **Enable autosave:** automatically save project every 5 minutes; the autosave interval can be set to whatever you like (**Autosave every** field). The total number of autosaves for a certain project can be limited (**Keep the last N autosaves** field; set is to 0 if you want not to limit the total number of autosaves for a certain project).
- **Autosave modified files only:** don't autosave if there were no modifications during the autosave period.
- **Beep on successful autosave:** play system beep sound when the map was autosaved successfully.
- **Use directory of the current file:** place autosaves to the same directory where "normal" map source is located; if you want to use a specific directory for autosaves, uncheck this option and set path to autosave directory in the **Autosave directory** field.

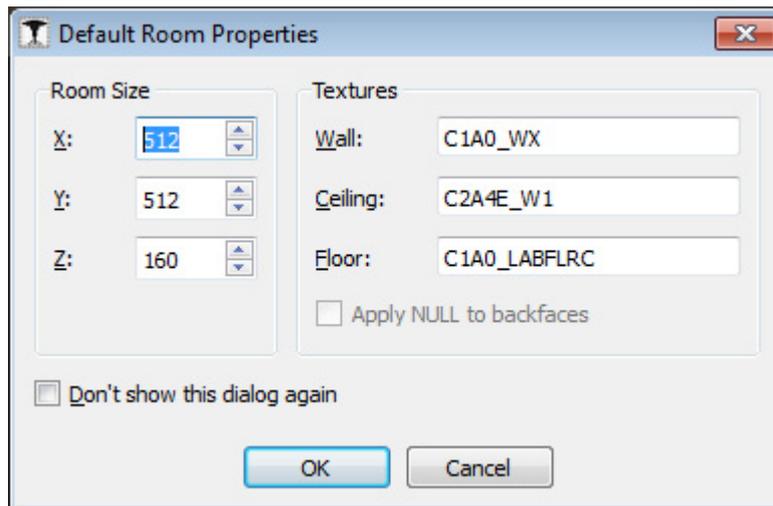
Steam Cloud

Note: these options are available in Steam version of J.A.C.K. only.

- **Enable remote storage:** master toggle for SteamCloud support.

- **Max. files:** maximum number of files to keep on the remote storage.
- **Store autosaves and backups:** store autosaves and backups on the remote storage along with "normal" map sources.
- **Automatically synchronize on startup:** always perform synchronization with SteamCloud on startup; if there are some files not up-to-date, it will bring the [Synchronization with SteamCloud](#) dialog.

Default Room Properties

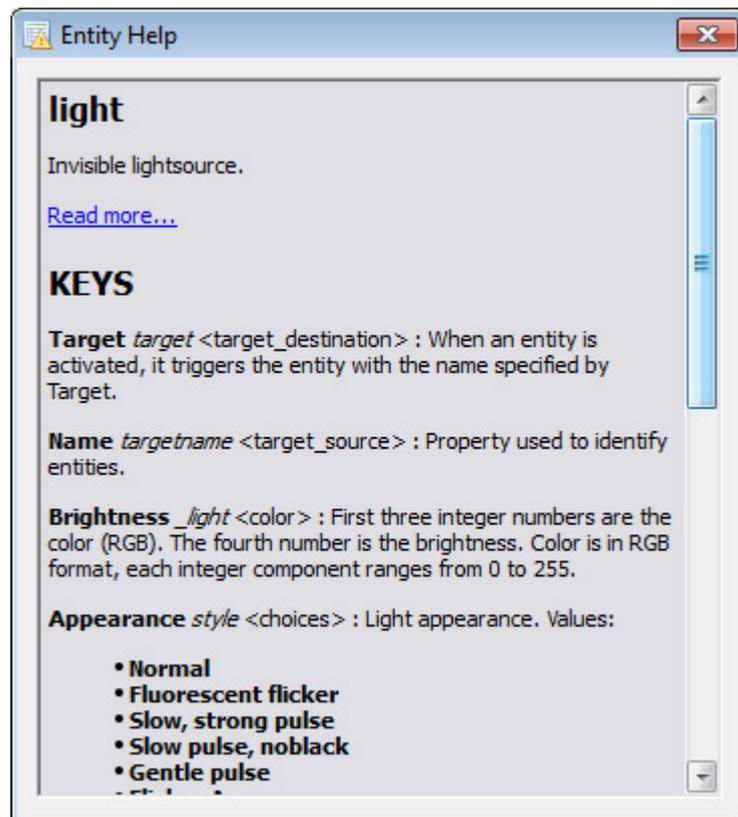


Hotkey: none

This dialog can be used to create a default room, which can be used as a good starting point to construct a level from scratch. You can specify room size in units, and textures to use for walls, ceiling, and floor. If your game profile defines a “null” texture, and there is such the texture amongst textures loaded for the current game configuration, you can check the **Apply NULL to backfaces** checkbox, so the backfaces will be covered with such the texture.

Don't show this dialog again: next time you create a new level, the dialog won't be displayed. To invoke it again, click the “New” button/menu item holding the **Shift** key.

Entity Help

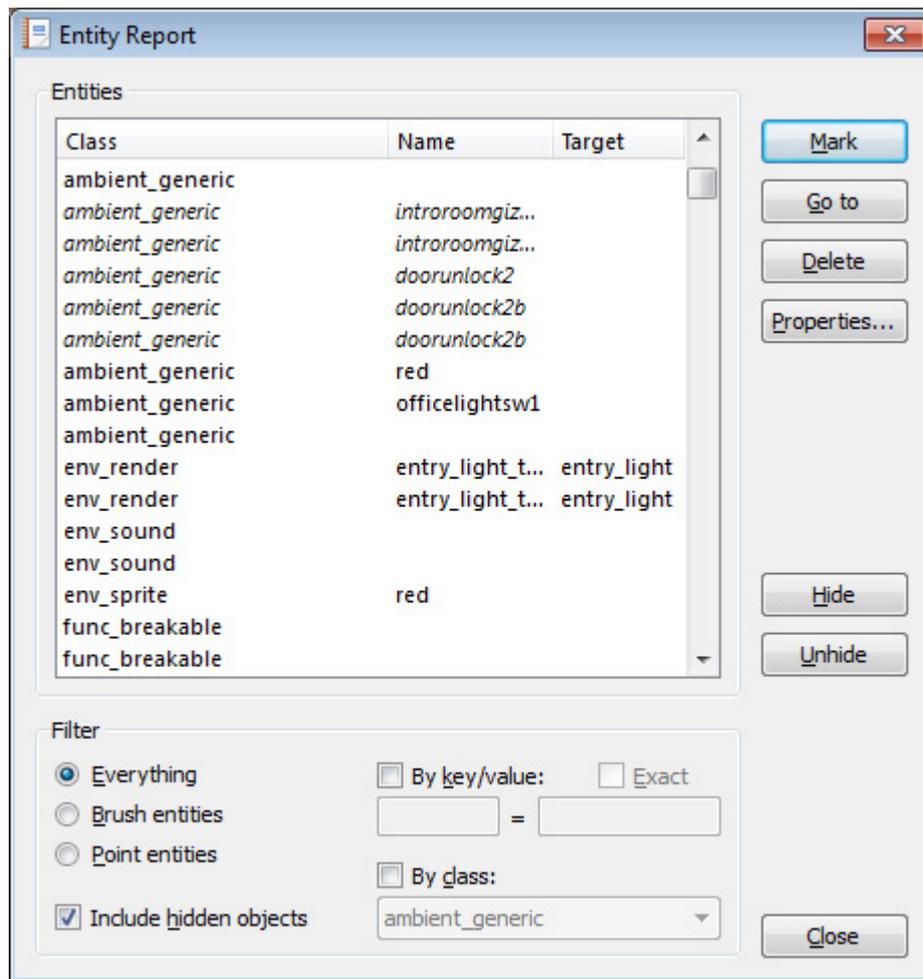


Hotkey: none

This dialog does nothing but shows detailed help for the current entity. This help is generated automatically based on the information provided in the FGD file.

Entity Report

This



Hotkey: none

The **Entity Report** can be used to manage your entities and do quick searches for specific entities, providing an easy alternative to searching through your map manually.

Entities

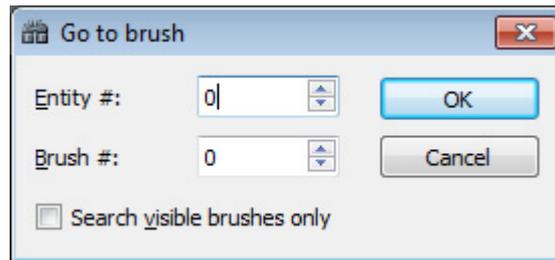
This box lists the entities in your level based on what is specified in the **Filter** section. The following keyvalues are shown: classname (entity type), targetname (the name of the entity), and target (the name of the entity this entity is targeted to). You can select either a single entity or multiple entities by Shift+clicking or Ctrl+clicking. With entities selected, you can use the six buttons to the right: Mark, Go to, Delete, Properties, Hide, and Unhide. Hidden entities are displayed in italic font.

Filter

Here you can specify filters so the only certain entities are displayed. First, you can choose whether all types of entities will be displayed, or only brush or point based entities. You can also decide whether currently hidden entities should be displayed.

When searching for a specific entity or group of entities, it will be convenient to use the search by key/value or search by class. While searching by key/values, entities with partial matches will be shown unless the **Exact** checkbox is checked.

Go to Brush



Hotkey: **Ctrl+Shift+G**

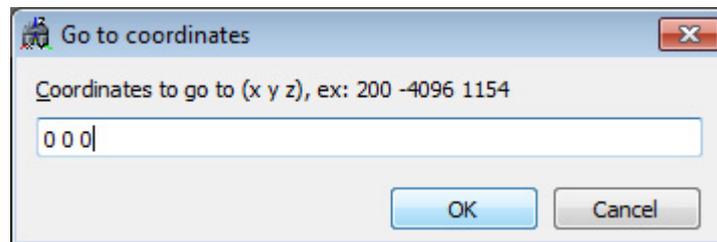
When you've got an invalid brush in your map, some compile tools will give you the id number of the brush that is bad. For example, Half-Life compile tools will give you something like this error:

```
Entity 0, Brush 4, Side 8: has a coplanar plane at (304, -384, 0), texture GENERIC99
```

To find the invalid brush in your map, you'd just bring up the **Go to Brush** dialog (by pressing **Ctrl+Shift+G**) and enter 0 as the entity number and 4 as the brush number, then press **OK**. The invalid brush will become selected and centered in the 2D views.

Note: if you've compiled or exported your map with the "Save visible objects only" setting enabled, you should enable the "**Search visible brushes only**" setting here as well.

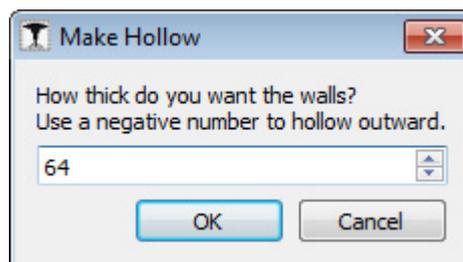
Go to Coordinates



Hotkey: none

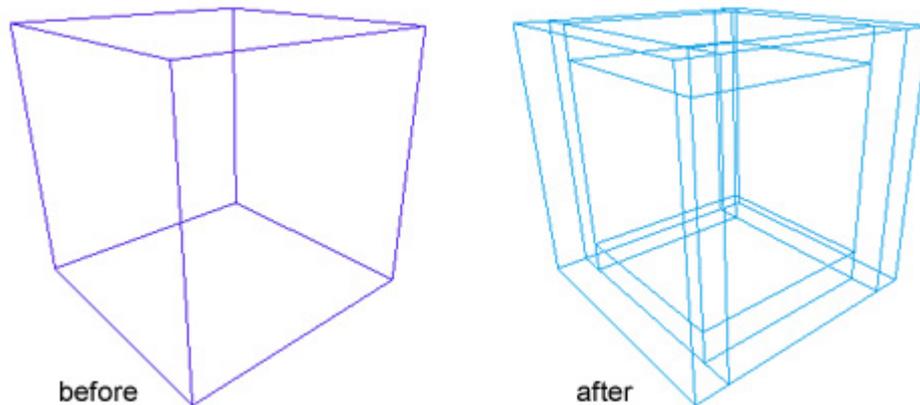
You can place camera in certain coordinates, to fix some errors as well. In the example described above, you can jump to coordinates 304 -384 0 (mind the absence of commas) and then search for an erroneous brush nearby. This is especially useful when you get problems related to space, and not particular brushes or entities (e.g. "Leaf portal saw into leaf" and "Too many light styles on face"), when only coordinates are spit by a compiler.

Hollow



Hotkey: **Ctrl+Shift+H**

The **Hollow** function allows you to take a block (or a set of blocks) and hollow it (them):

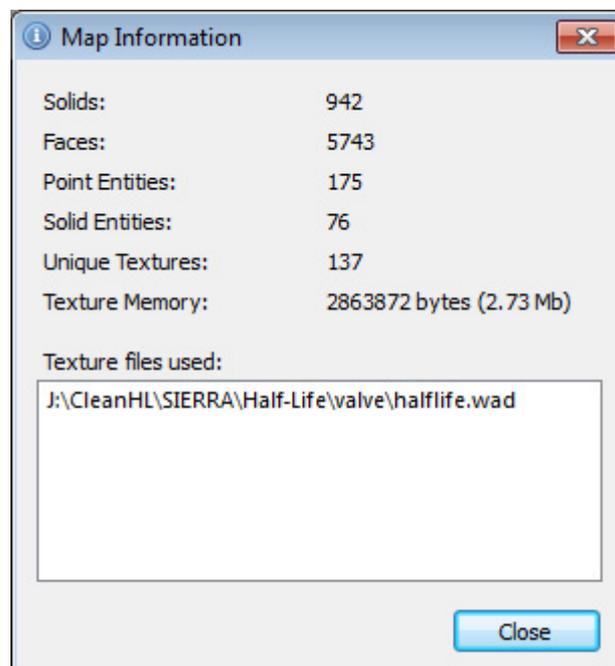


To hollow outward (i.e. to make the block the inside area of the hollow cube), use a negative number for the wall thickness.

If you have multiple blocks selected, the message box will appear to confirm the operation.



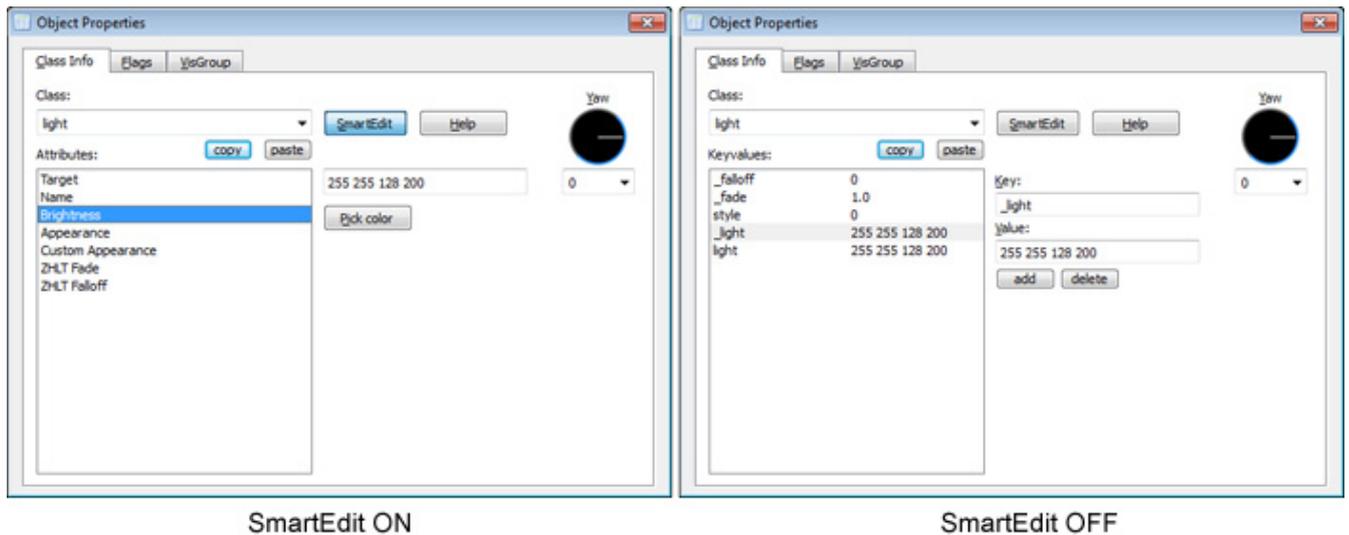
Map Information



Hotkey: none

This dialog gives some statistical information about the current map.

Object Properties



Hotkey: **Alt+Enter**

The **Object Properties** dialog box is the primary method of modifying entity properties. It also allows you to adjust VisGroup properties for one or many brushes.

Class Info

You can modify entity properties in the **Class Info** tab. Class Info has two modes, a regular editing mode and **SmartEdit** (which is toggled on and off by the SmartEdit button). When you use the SmartEdit mode, all of the entity's key (entity variable) descriptions are shown to you. If you are already familiar with the entity properties, you may find it more convenient to turn this mode off and manually insert the keyvalues. This section is only available for entities.

The angle control is comprised of two parts: the **Yaw** text box and the compass. These two allow you to accomplish the same thing: setting an angle between 0 and 359. The Yaw text box also allows you to choose “Up” and “Down” which corresponds to Quake-style angles -1 and -2.

The **Help** button invokes Entity Help dialog for the current entity class.

For some entity properties (those that require a path/filename of a sprite, sound, or model) you will have a small button beside their property when you click on them. Clicking on this button will allow you to browse through the appropriate game directory for the sprite, sound, or model. This behavior is controlled by the game data file.

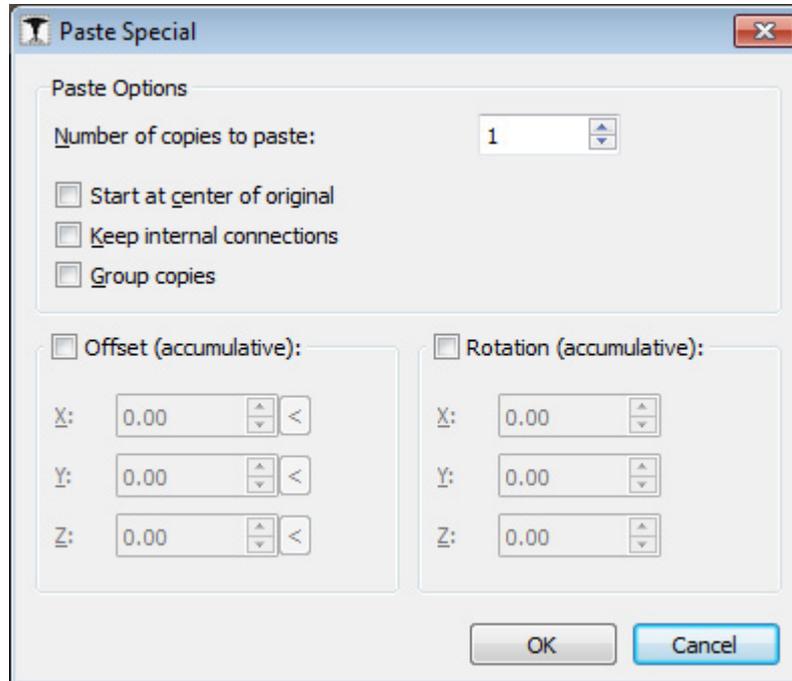
Flags

You will find listed here all the available flags for an entity. This section is also only available when editing an entity's properties. A flag allows you to toggle on or off specific features of an entity. The numeric values of the flags are shown as tool tips, when you hold mouse cursor over a checkbox for a while.

VisGroup

This section allows you to assign the selected object to a VisGroup or remove it from one. You can also bring up the Object Groups dialog from here. For more information, see [Grouping and VisGrouping](#).

Paste Special



Hotkey: none

The **Paste Special** dialog box can be used to insert multiple copies (**Number of copies to paste**) of the clipboard contents, with optional offsets and rotations, and some additional options.

Paste options are:

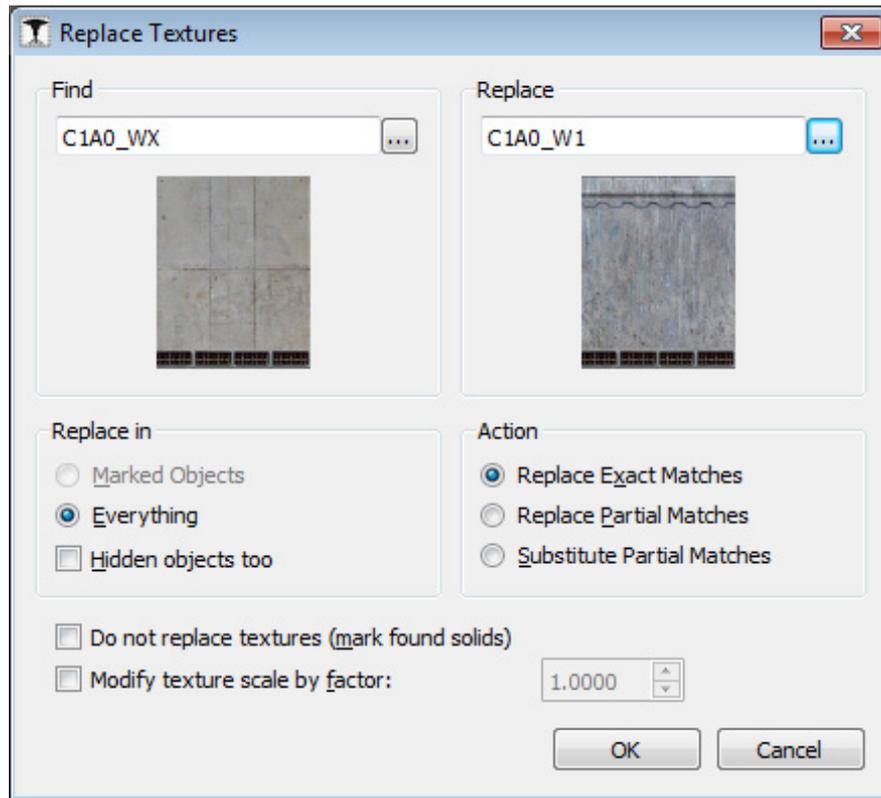
Start at center of original: paste coordinates will be the same as source object's coordinates.

Keep internal connections: targetname-target links will be maintained unique; e.g. this allows pasting of multiple func_buttons targeted to certain lights; each pasted button will target its own light.

Group copies: each pasted copy of clipboard contents will be added to a separate group.

Offset and rotation: accumulative offsets (in units) and rotations (in degrees) applied to pasted copies. E.g. in order to create stairs, start with a single step, copy it to clipboard (**Ctrl+C**) and then paste-special a number of copies with certain horizontal and vertical offsets. The "<" buttons will place a corresponding dimension of the bounding box to the offset field.

Replace Textures



Hotkey: none

Find/Replace

These text boxes and texture pictures show the texture to be found in the map, and the new, replacement texture. You can browse through the entire texture library to choose the “find” texture, the “replace” texture, or both.

Replace in

The Replace in section lets you specify what should and shouldn't get replaced. You can replace everything, or only marked objects, and you can choose to include or exclude hidden objects.

Action

This lets you choose which search-and-replace method you would like to use. **Replace exact matches** means that exact “Find” texture names will be replaced. **Replace partial matches** means that “Find” is not a texture name, but a part of texture name to replace.

Substitute partial matches is a more complex tool; when this is enabled, you can search for a specific part of texture name, and substitute that part with something else. For example, if you have a bunch of slime textures (“slimedoor”, “slimefloor” etc.), and you want to change them to “rustdoor”, “rustfloor” etc., you can enable this, search for “slime” and replace with “rust”.

Do not replace textures (mark found solids)

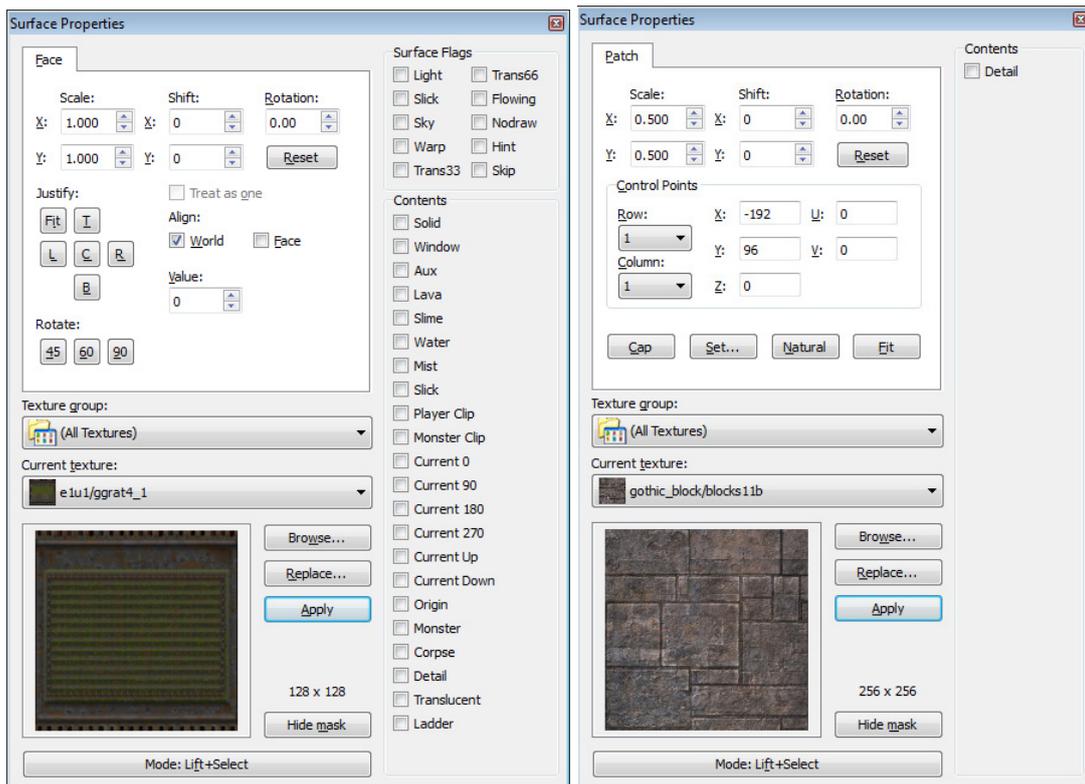
Use this feature to simply mark (select) brushes (or faces, in texture application mode) containing the given texture, instead of replacing them with another texture. This is extremely useful if you would

like to modify the texture attributes of every texture of certain type, but don't want to select each one manually (while you are in the Face Properties dialog).

Modify texture scale by factor

Sometimes you need to replace a smaller texture with a larger one, or vice versa. To keep texture aligned correctly, you can rescale them by a certain value, actually a ratio between new and original texture sizes.

Surface Properties



Hotkey: **Shift+A**

The **Surface Properties** dialog allows you to manipulate the texture properties of selected brush faces. To select a brush face, click on it in the 3D view with the left mouse button. To select multiple faces, hold down the **Ctrl** key while you do this. You can also select multiple brushes *before* bringing up the Texture Application tool.

A special feature of the **Surface Properties** tool is that of being able to apply the current texture to a brush face by **right-clicking** on that face. In order to make the textures match up from one face to the next, first align the texture on one of the faces, lift the texture from that face (**left-click** by default), then **Alt+right-click** on the other faces one by one. The texture should flow continuously across all the faces. You can also move the texture by mouse in 3D views: **left-click** the face holding **Alt**, and then drag the texture in the desired direction (not applicable to curves).

The dialog may look different, depending on the current game profile, and the objects selected. Left screenshot shows a dialog displayed in Quake II profile, with a normal brush selected; right screenshot shows a dialog displayed in Quake III profile, with a curve selected.

The following options are available in all profiles:

Scale (X/Y)

You can modify the scale of a texture to shrink or enlarge it. Numbers less than 1 shrink, and numbers greater than 1 enlarge. You can use negative values to mirror the texture in one or both planes.

Shift (X/Y)

This simply allows you to shift the texture around in the X and/or Y planes.

Rotation

Rotation allows you to rotate the texture on a brush face.

Reset

This button resets scale, shift and rotation to the defaults.

Texture Group

You can select which group of textures you want to use.

Current Texture

The name of the current texture is displayed here, along with a thumbnail picture of the texture.

Browse

This button opens the [Texture Browser](#).

Replace

This button brings the [Replace Textures](#) dialog.

Apply

This button applies current texture and values to the selection.

Hide Mask

If this button is toggled on, the selected brush face(s) will not appear as selected (with a reddish hue). This may make it easier to perform some texture manipulations.

Mode

This lets you select what feature you want the left mouse button to be. There are different options available:

- **Lift+Select:** clicking on a brush face or curve in the 3D view will select that face and make it's texture the current texture.
- **Lift:** clicking on a brush face or curve will make it's texture the current texture without selecting the face.

- **Select**: clicking on a brush face or curve will select that face/curve without changing the current texture.
- **Apply (texture only)**: clicking on a brush face or curve will apply the current texture to that face/curve without changing any of its other properties.
- **Apply (texture + values)**: clicking on a brush face or curve will apply the current texture and texture properties.
- **Apply (texture + axes + values)**: clicking on a brush face or curve will apply the current texture, texture properties, and texture orientation (i.e. texture axes).
- **NULL to Selected**: in game profiles which support the null texture, clicking on a brush face or curve will apply that texture.
- **NULL to Unselected**: in game profiles which support the null texture, clicking on a brush face will apply that texture to all brush faces *except* the clicked one.
- **Align to View**: this is like projecting a texture from the camera face, as if you were a slide projector and the texture is a slide. It's useful for painting a texture onto a cliff or other irregular things.

Surface Flags / Contents

In some game profiles, surface flags can be specified on a per-face basis. The same applies to contents, which are actually per-brush (i.e. must be set identical to all the faces of a single brush). Surface flags and contents are game-specific; please refer to the game documentation to find out meanings of certain flags.

Face-specific options include:

Justify

There are six buttons you can press here: **L** (left), **R** (right), **Fit** (fit to face), **T** (top), **B** (bottom), and **C** (center on face). These allow you to quickly and easily align a texture on a face. Click the **Fit** button holding **Shift** to bring up the **Fit Texture** dialog, which allows specifying a number of horizontal and vertical texture repeats (by default, fitting means no repeats, i.e. 1:1 ratio).

Treat as one

If you have multiple faces selected, you can check this option to treat all the faces as one single face.

Align: World and Face

These checkboxes allow you to set the texture alignment style for the selected faces. *World* alignment causes textures to be aligned according to world coordinates (if you're familiar with old versions of Valve Hammer Editor/Worldcraft, this was the standard behavior). *Face* alignment uses the coordinates of the face as a basis for alignment.

Value

In some game profiles, the **Value** field is available; it defines amount of light emitted from this surface, which is effectively transformed to an “area” light source.

Rotate

There are three buttons you can press to quickly rotate the texture by 45, 60, or 90 degrees. Hold Shift when clicking these buttons to rotate in the opposite direction.

Curve-specific options include:

Control Points

You can manually edit control points of the curve, by selecting row and column indices, and then modifying vertex coordinates (XYZ) and texture coordinates (UV).

Cap

This texturing feature is not yet implemented.

Set...

This brings up the **Fit Texture** dialog, which allows specifying a number of horizontal and vertical texture repeats along the curve.

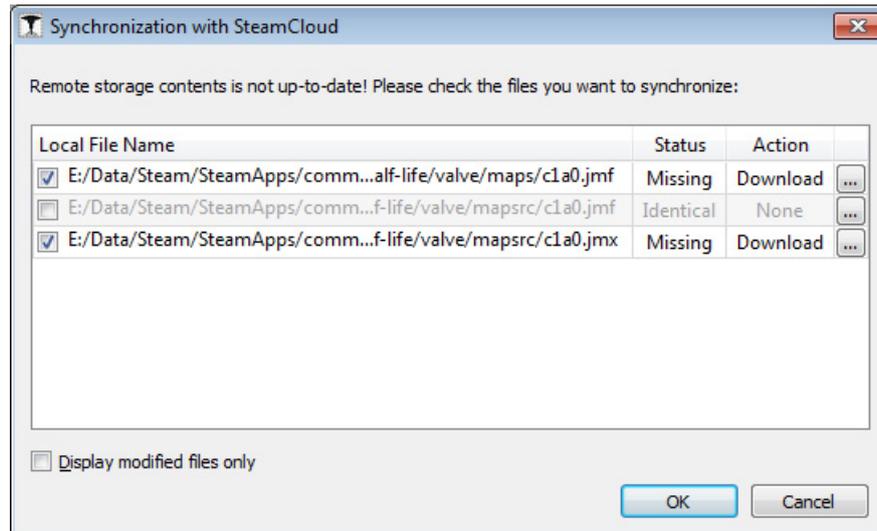
Natural

Performs texture naturalization along the curve; this means that texture coordinates are calculated depending on actual edge length. As the result, the texture will look “natural” without unnecessary extensions and shrinks.

Fit

Fits the texture to the curve, which is actually the same as executing the **Set...** command with 1:1 repeat ratio.

Synchronization with SteamCloud



Hotkey: **F5**

Note: this dialog exists in the Steam version of J.A.C.K. only!

The **Synchronization with SteamCloud** dialog is the only way to interact with the remote cloud storage (SteamCloud). Here you can download missing files, upload new versions (if they were modified not in J.A.C.K. or while the SteamCloud was temporarily disabled), and delete files from SteamCloud.

The file list shows map files that are currently under the "source control". **Status** describes what's wrong with them (if any):

- **Identical:** it's OK (local and remote files are identical).
- **Missing:** a local file doesn't exist anymore.
- **Older:** a local file is older than a remote file.
- **Newer:** a local file is newer than a remote file.

Action describes what type of synchronization to perform:

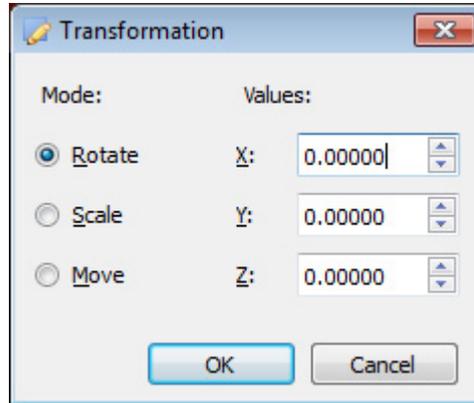
- **None:** do nothing.
- **Download:** download a remote file, and replace a local file with the remote file (if the local file exists).
- **Upload:** upload a local file to the remote storage.

Actions will be performed only on checked files, so uncheck the files you want to keep intact. Press the ... (three dots) button to open there context menu, where you have a delete command. That command doesn't delete the local file; only the remote file will be deleted, and the local file will be removed from the "source control".

Display modified files only: check this to remove files with an "Identical" status from the file list.

Note that all the actions are batched, and they are not performed until you press **OK**. Press it to confirm the action list, and to actually perform synchronization. Press **Cancel** (or **Esc**) to close the dialog without any synchronization.

Transformation



Hotkey: **Ctrl+M**

The **Transformation** dialog gives you precise control over a selected object. You have the option to rotate, scale, or move the selected object in any of the X, Y, or Z planes. Values entered are in units, except in the case of rotation, where degrees are used.

Note: this is the only way to move objects by a fractional number of units, including numbers less than 1 (minimum grid size). I.e. this can be used (although not recommended) to place objects off the 1-unit grid.

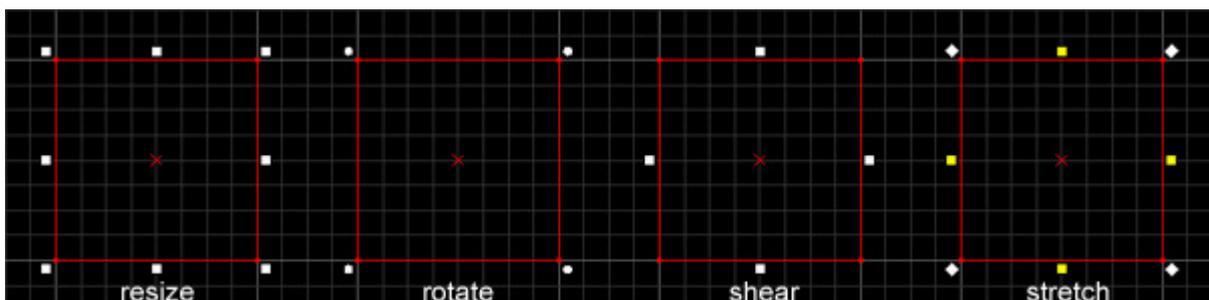
5.3. Toolbars

Map Tools

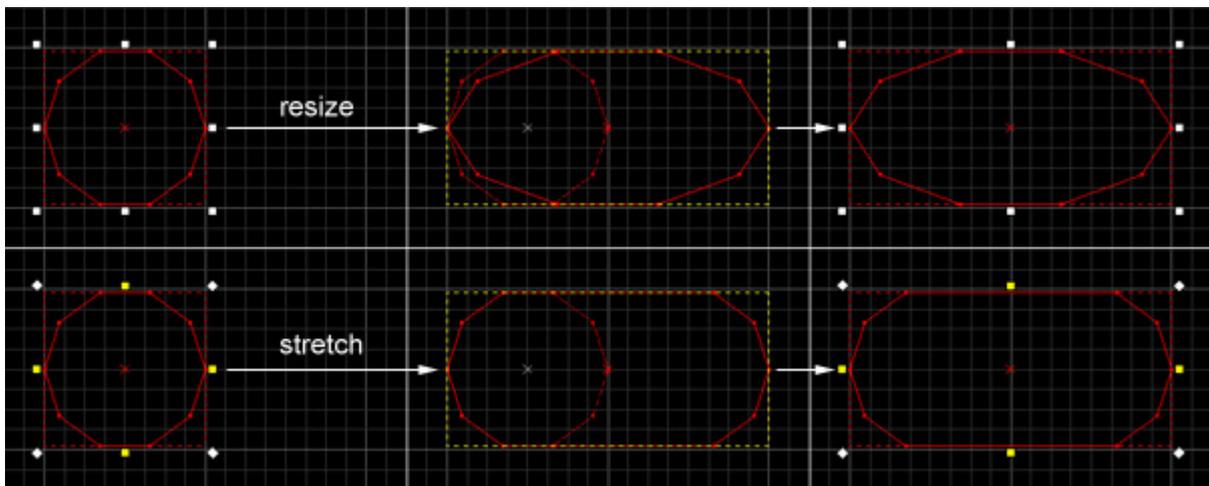


The Map Tools toolbar lets you switch between the different editing modes. This toolbar can safely be closed as each button also links to a hotkey.

 **Selection Tool (Shift+S).** The selection tool allows you to accomplish a number of things. To simply select an object, click once. Continue clicking on the selected area to cycle through the four basic transformation modes (resize, rotate, shear, and stretch).



- **Resize** mode allows you to grab any corner or edge of the selected object and resize it by pulling it to the desired size.
- In **rotate** mode, you can grab any corner of a selected object and pull it around to rotate the object into any position you wish. If you have the rotations set to 15 degrees by default (this is set in the 2D Views options), this can be overridden by holding **Shift** while you rotate the object.
- **Shear** mode lets drag the edges of an object parallel to its surface. For example, if you grab the top edge of a selected object while in shear mode, you would be able to drag that edge right or left.
- **Stretch** mode is a bit tricky to describe. It is very similar to **resize** mode, but instead of resizing the whole object, it attempts to extrude *only parts* of the objects *half-way* to the edge being stretched. You can see the difference between **resize** and **stretch** at the figure:



Tip: You can clone a selected object by holding down **Shift** and click-dragging the object. This can be done regardless of what selection mode the object is in (resize, rotate, or shear). See [Creating Solids](#) for more information.

Note: While in selection mode, you can select multiple objects by dragging a selection box around them and pressing **Enter**. If **Auto-Selection** is toggled on, pressing **Enter** is not necessary.



Magnify (Shift+G). This tool allows you to increase the magnification factor of the 2D views by clicking left mouse button. Zoom back out again by clicking the right mouse button. You can also zoom in and out by pressing the number keys while the mouse cursor is in a 2D view, or by using the + and - keys.



Camera (Shift+C). This tool allows you to place and modify cameras within your level. While in camera mode, hold **Shift** and left-click-drag a line to create the camera. A line with a dot (the view point) will be drawn in the 2D views, where you can adjust it until it is pointing in the direction you'd like. To place more than one camera in a level, just repeat the above steps.

Cameras are good for helping you visualize the creation of your level. They are especially useful when using the **Vertex Manipulation** tool as they allow you to see exactly what you are doing to a brush. You will find multiple cameras especially useful if you are working on several areas of a level at once. Placing a new camera in an area negates the need to search for that area when you want to work on it again.

When you're in camera mode, you can cycle through all available cameras by pressing the **PageUp** or **PageDown** keys. To delete the current camera, press **Delete**. More information can be found in [The 3D and 2D Views](#).



Entity Creation Tool (Shift+E). This tool allows you to place point-based entities in a map. (Point-based entities are those entities which exist only at a point, and do not rely on a brush for their effect) Once selected, the [New Objects](#) dialog will contain a list of entities from which to choose. Click on one of the 2D views and press Enter to place the entity. If Snap To Grid is enabled, the new entities will be snapped to the grid.

Note: in order to place a brush-based entity (a func_door for example) you must first create the brush with the Block Tool (see below), then select it, and press Ctrl+T. You will be given a list of entities to choose from which will be different than those shown with the Entity Tool.



Block Creation Tool (Shift+B). This is the basic creation tool available to you. It allows you to create any of the different types of primitive shapes. For more information, see [Creating Solids](#).



Texture Application Mode (Shift+A). This toggles the [Surface Properties](#) dialog on and off. This dialog allows you to edit the texture properties of individual brush faces. For more information, see [Texturing Solids](#).



Apply Current Texture to Selection (Shift+T). When pressed, the current texture is applied to every face of the selected brushes. For more information, see [Texturing Solids](#).



Apply Decals (Shift+D). A decal is a texture that can be placed on top of another texture. Common examples include scorch marks or bullet holes. Use the left mouse button to place a decal in the 3D window. Decal textures can be viewed and selected in the **Textures** docking panel. Half-Life's decal textures are designated by the use of a { as the first character in the texture name.

Note: not all the game profiles support decals; currently, only Half-Life supports them.



Clipping Tool (Shift+X). Clipping allows you to slice the currently selected brush. You have the option of only splitting the brush, or actually slicing a piece right off. For more info, see [Reshaping Solids](#).



Vertex Manipulation (Shift+V). Vertex manipulation gives you complete control over the shape of a brush. You can reshape the brush by manipulating individual vertices and brush edges. For more information, see [Reshaping Solids](#).



Path Tool (Shift+P). This tool greatly simplifies the creation of paths for trains and monsters. It allows you to create dynamic paths that you can add and delete nodes to. Refer to the [Path Tool](#) chapter for the details.

Map File



This toolbar allows performing common file operations.

1. **Create a new document.** Creates a completely new project; the Default Room dialog appears. If you selected not to show that dialog again, hold Shift when clicking to bring this dialog up again.
2. **Open an existing document.** Opens an existing project from disk. You can select a proper format in the Open File dialog, for MAP files, where many different formats share the same suffix.
3. **Save the active document.** Saves the previously saved project to disk. If the project is not yet saved, the Save File dialog appears, as if you clicked **Save As...** command in the File menu.
4. **Export level again to the same format.** Exports the previously exported project to the MAP file which can be then compiled by an external compiler. If the project is not yet exported, the Save File dialog appears, as if you clicked **Export Map...** command in the File menu.

Map View



This toolbar controls some of the view settings for J.A.C.K.

1. **Toggle grid, Toggle 3D-grid.** These toggle the grid on and off in the 2D and 3D views.
2. **Lower/raise the size of the grid.** These decrease and increase the grid size. The [and] hotkeys can also be used for this purpose.
3. **Load default window state, save current window state as default.** These buttons are used to save and restore window configurations when the independent window configuration is being used. Independent window usage can be enabled or disabled in the [General options](#).
4. **Show/hide texture tiling UV grid.** This button toggles display of the UV-grid, i.e. texture image borders, in 3D views.

Map Transform



This toolbar facilitates access to some useful transform commands.

1. **Rotate, scale and/or move** the selected objects.
2. **Align the selected objects** along the topmost/leftmost/topmost/bottom-most side.
3. **Flip the selected objects** horizontally/vertically.
4. **Rotate the selected objects** anti-clockwise/clockwise.

Map Operations



The Map Operations toolbar allows you to perform a number of operations on objects in your map as well as setting default behavior for some mapping properties.

1. **Carve using the selected objects.** When the Carve button is pressed, the selected brush will subtract its shape from the non-selected brushes around it. See the [Carving](#) chapter for the details.
2. **Merge the selected objects.** When the Merge button is pressed, the selected brushes forming the convex hull will be merged together. See the [Brush merging](#) chapter for the details.
3. **Group/Ungroup.** The first command binds two or more objects together so they may be acted upon simultaneously; the second releases the top-level group(s) previously created. This is not the same as VisGrouping. The two functions differ in that grouping provides a way of physically binding a group of objects together, while VisGrouping enables you to organize objects into a group that may still be worked upon as separate objects. For more information, see [Grouping and VisGrouping](#).
4. **Toggle group ignore.** Ignore Groups allows you to modify individual brushes that are part of a group or an entity by temporarily disabling "groups".
5. **Hide Selected/Unselected, Show All.** Along with the [Filter Control](#) docking panel, these buttons are used to control what does and doesn't appear in the 2D and 3D space of J.A.C.K.
6. **Cut, Copy, Paste.** These buttons follow the standard interface for cut/copy/paste.
7. **Toggle cordon state, Edit cordon boundaries.** These buttons allow you to "cordon off" an area of your map. When the cordon state is toggled on, a thick red line will appear around that area. When you compile, only the inside of that red area will be compiled, and a box will be placed around the area to seal leaks. This is extremely useful when you are working on a large level and want to look at changes made in only a small area. Rather than wait a considerable amount of time for the entire map to compile, you can selectively compile only a small area by using the Cordon functions.

Note: be sure to place a temporary player start within the bounds of the cordoned area before you test compile that section.

8. **Select by handles only.** When this option is toggled on, you will only be able to select objects by their center handles in the 2D view. To select objects by their center handles in 3D view, you

- must also be in wireframe mode. This is useful when there are several brushes overlapping in the 2D view, making the standard selection method (clicking on brush lines) awkward and inaccurate.
9. **Toggle 2D auto-selection by selection frame.** When 2D auto-selection is toggled on, you can select multiple objects by simply clicking your mouse button and dragging a selection box around the objects. When this option is toggled off, you can also select multiple objects by dragging a selection box around the objects and then pressing **Enter**.
 10. **Toggle 3D auto-selection by mouse dragging.** When 3D auto-selection is toggled on, you can click-and-drag left mouse button to select everything under the cursor (solids, entities, faces, etc.) When this option is toggled off, you have to click every single object holding Ctrl to select multiple objects.
 11. **Texture Lock (movements+rotations, scale, UV-lock).** These buttons toggle different types of texture lock mode on and off. Textures will then stay correct when you move or rotate a brush (the first button), when you scale the brush (the first *and* the second button), or perform even more complicated operations, such as shear or vertex manipulation (the third button).
 12. **Align to World/Face.** This button toggles the *default* texture alignment style between world alignment and face alignment. World alignment will align textures according to the world grid coordinates. Face alignment will align textures according to the brush face.
 13. **Toggle solid validity restrictions.** This button toggles invalid solid structure creation prevention during vertex manipulation. When the button is *not* pressed, the restrictions are in effect; otherwise, there are no restrictions at all.
 14. **Toggle automatic coplanar face merging.** This button toggles automatic coplanar face merging during vertex manipulation. Coplanar faces are two or more adjacent faces on the same plane, and this is usually a severe map error. By default (the button is *not* pressed) J.A.C.K. automatically merges such faces wherever possible. However, you sometimes need time to work on a single solid, entering and exiting vertex manipulation several times, and this can be a useful option.
 15. **Run the map in the game.** This button brings up the current compile dialog. After successful compilation, the game on the current level may be started (depends on what you selected in the Run Map dialog). For more information on compiling you map, see [Compiling and Running Your Level](#).

Map Render

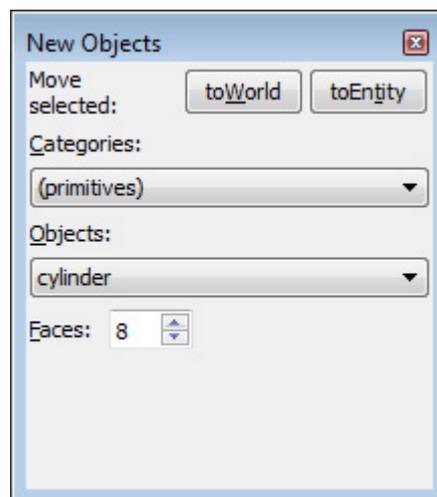


The Map Render toolbar allows you to customize rendition in 2D and/or 3D views.

1. **Toggle transparency of special textures.** This button toggles transparency of some utility textures (e.g. trigger, clip, etc) in 3D views.

2. **Toggle transparency of entities with non-standard rendermodes.** This button toggles visualization of entities' render modes. Note that not all the games support rendermodes. They are actually applicable for Half-Life and Quake (if you use some modifications, such as Quoth).
3. **Show/hide connections between Targets and their Triggers.** This enables display of targetname-target links as arrows in 2D and 3D views. For more information about target names and targets, refer to the [Names and Targets](#) chapter.
4. **Show/hide models in 2D.** This button toggles wireframe model rendition in 2D views. When this option is turned off, models are displayed as bounding boxes, and this is a Valve Hammer Editor/Worldcraft style behavior; otherwise, you'll be able to see exact meshes.
5. **Show dynamic skybox in 3D.** This enables a dynamic sky box instead of simple sky polygons. Such display is very close to what you see in game. Dynamic sky box is available only if the proper sky textures are loaded.
6. **Animate textures and effects in 3D.** This button toggles WYSIWYG display of textures and effects in 3D views. The available effects depend in the game profile. E.g., in Half-Life profile you'll see texture animations and scrolling; in Quake III, "shaders" will be fully visualized, including deformations, color and texture coordinate animations, etc.
7. **Show/hide nodraw/hint/clip/trigger faces in 3D.** These buttons toggle visibility of faces painted with special textures, such as nodraw/null, hint/skip, clip/aiclip, and trigger. This doesn't affect the presence of such faces in the output map file.

Object Bar



Hotkey: **F6**

The **New Objects** docking panel controls all the facets of creation within J.A.C.K. Entities, brushes, curves — the creation of all this can be controlled from here.

toWorld (Ctrl+W): this will remove any entity settings of the selected solid or group of solids.

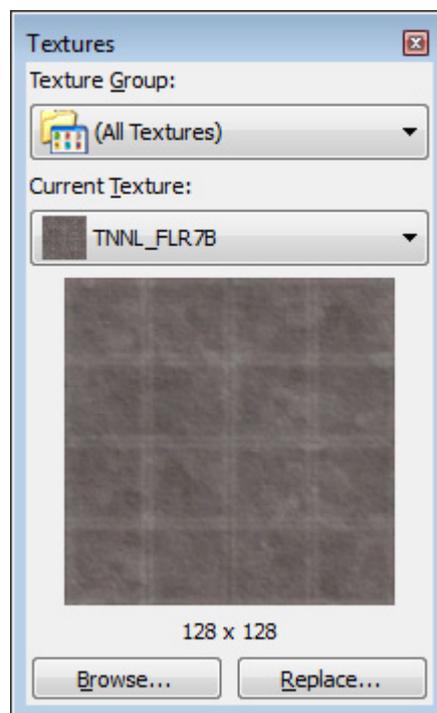
toEntity (Ctrl+T): this will attach the currently selected solid or group of solids to an entity; you will be offered to select an entity to keep, if multiple entities are selected. Hold **Shift** while clicking this button, if you have multiple solids selected, and want to create a separate entity for each solid.

Categories: when the **Entity Tool** is active, no categories are available. With **Block Tool** selected, there will be at least one category available: the "(primitives)", which contain J.A.C.K.'s seven basic solid shapes: block, cylinder, spike, wedge, sphere, rock, and the arch tool. If the game profile supports curves, then the "(curves)" section will also be listed here, which contains many curve primitives as well.

Objects: this list contains all objects that belong to the chosen category. For entities, the list contains all entity classes available. For solids, if the "(primitives)" category has been selected, then this list contains the seven types of primitives. If the "(curves)" category has been selected, then this list contains types of curve primitives.

Faces: this text box becomes active if the **Block Tool** is active, and one of the four primitives (cylinder, spike, sphere, rock) is the current solid type. The value (which can be changed by either the keyboard or clicking on the little arrows) represents the number of side faces the cylinder/spike/sphere/rock will be created with.

Texture Bar



Hotkey: **F7**

The **Textures** docking panel is used to select the current texture.

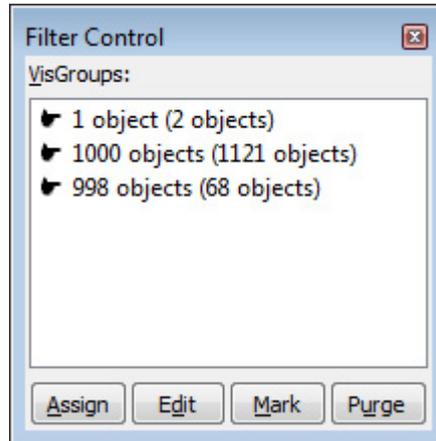
Texture Group: the texture group pull-down menu lets you choose either "(All Textures)", or individual texture packages. This affects which texture package will be active when you **Browse**.

Current Texture: this lists the name of the current texture and below that, a small picture of the texture. The pull-down menu lists the most recent texture choices. Under the texture image, the original texture size is shown (width x height).

Browse: this brings up the [Texture Browser](#).

Replace: this will bring up the [Replace Textures](#) dialog.

Filter Control



Hotkey: **F8**

All VisGroups will be listed in this docking panel.

VisGrouping is a powerful tool for limiting visible objects in the map. This allows for much easier editing on large levels but does require that you use the JMF file type for saving your maps since the MAP format doesn't include the VisGroup information. Note that RMF and VMF formats support VisGroups as well.

VisGroups are created using the **Edit Groups...** dialog. When a VisGroup is created, it is inserted into the **Filter Control** panel with a suffix of "(n objects)", where n is the number of objects in the VisGroup. The name can be changed by clicking on the group, then clicking the **Edit** button.

There are a number of buttons that can be used in the **Filter Control** dialog:

- **Assign:** this will cause selected objects to be added to currently selected VisGroup.
- **Edit:** this allows you to edit the name and color of a VisGroup, as well as add and remove new groups.
- **Mark:** this causes selected VisGroup to be highlighted in the 2D and 3D views.
- **Purge:** this will remove any groups from the VisGroup window that no longer contain any objects.

Another feature of the **Filter Control** window is the ability to merge two VisGroups. This is done by dragging one group on top of a second. J.A.C.K. will ask if you want to combine the two groups, and if you do, they will be merged and the first group will take on the VisGroup properties (name, color) of the second group.

For more information on VisGroups, see [Grouping and VisGrouping](#).

Status Bar



The status bar contains a number of important pieces of information. Each bit of information has its own little area on the bar. The areas are as follows.

Selection information. The first bit of information on the status bar is about what is currently selected. If you've got a non-entity brush selected, you will see something like "solid with 6 faces". If you've got an entity selected, you will see something like "func_breakable" or "name1 - func_breakable" (the second example "name1 - func_breakable" - will be shown if you've selected an entity that has a name).

Coordinates. Follow the selection information, you'll see something like "@32, 144". These are the coordinates of the mouse cursor in whichever 2D view it's currently in.

Selection size/center information. When you select a brush or entity, its size and center coordinates will be listed here. You'll see the width, length, and height values listed as something like "32w 128l 64h", and xyz coordinates of the selection center appended as "@ (0 0 32)".

Zoom size. If a 2D view is currently in focus, its zoom value will be shown next.

Grid information. The last box displays something like "Snap: On Grid: 16". This means that Snap To Grid is enabled (**Shift+W** to toggle it) and the current grid is in 16 unit increments (Use [and] to decrease and increase the size).

5.4. Hotkey Reference

- **Alt+E:** Scale vertices (Vertex Manipulation)
- **Alt+P:** Check map for problems
- **Alt+F3:** View message window
- **Alt+Backspace:** Undo the last action
- **Alt+Enter:** Edit object properties
- **Alt+Shift+Backspace:** Redo the previously undone action
- **Ctrl+3:** Center the 3D views on the selected objects
- **Ctrl+4:** Autosize the 4 views in a map window
- **Ctrl+A:** Select all objects
- **Ctrl+B:** Snap the selected objects to the grid (without altering size/relativity)
- **Ctrl+C:** Copy the selection and put it on the Clipboard
- **Ctrl+D:** Hide all detail brushes
- **Ctrl+E:** Center the 2D views on the selected objects

- **Ctrl+F**: Split face (Vertex Manipulation)
- **Ctrl+G**: Create an associative group with all the selected objects
- **Ctrl+H**: Hide the UNselected objects temporarily
- **Ctrl+I**: Flip the selected objects vertically
- **Ctrl+K**: Make second selected entity a target of the first selected entity
- **Ctrl+L**: Flip the selected objects horizontally
- **Ctrl+M**: Rotate, scale and/or move the selected objects
- **Ctrl+N**: Create a new document
- **Ctrl+O**: Open an existing document
- **Ctrl+Q**: Unselect all selected objects
- **Ctrl+R**: Rotate the selected objects anti-clockwise
- **Ctrl+S**: Save the active document
- **Ctrl+T**: Create a new entity using the selected solids
- **Ctrl+U**: Make each of the selected solids a group of hollow objects
- **Ctrl+V**: Insert Clipboard contents
- **Ctrl+W**: Detach the selected solids from their entity
- **Ctrl+X**: Cut the selection and put it on the Clipboard
- **Ctrl+Y**: Redo the previously undone action
- **Ctrl+Z**: Undo the last action
- **Ctrl+F2**: Automatic creation of game configurations
- **Ctrl+F4**: Close the active document
- **Ctrl+F6**: Switch to the next document window
- **Ctrl+Ins**: Copy the selection and put it on the Clipboard
- **Ctrl+Tab**: Switch to the next document window
- **Shift+A**: Texture Application Mode
- **Shift+B**: Block Creation Tool
- **Shift+C**: Camera
- **Shift+D**: Apply Decals
- **Shift+E**: Entity Creation Tool
- **Shift+G**: Magnify
- **Shift+I**: Invert selection of all objects
- **Shift+L**: Toggle texture locking for movements and rotations
- **Shift+P**: Path Tool
- **Shift+Q**: Unselect all selected objects

- **Shift+R**: Toggle grid
- **Shift+S**: Selection Tool
- **Shift+T**: Apply Current Texture to Selection
- **Shift+V**: Vertex Manipulation
- **Shift+W**: Toggle automatic snapping of vertices to grid
- **Shift+X**: Clipping Tool
- **Shift+Z**: Maximize current view
- **Shift+F1**: Set mode "Lift + Select" (Texture Tool)
- **Shift+F2**: Set mode "Lift" (Texture Tool)
- **Shift+F3**: Set mode "Select" (Texture Tool)
- **Shift+F4**: Set mode "Apply (texture only)" (Texture Tool)
- **Shift+F5**: Set mode "Apply (texture + values)" (Texture Tool)
- **Shift+F6**: Set mode "Apply (texture + axes + values)" (Texture Tool)
- **Shift+F7**: Set mode "NULL to Selected" (Texture Tool)
- **Shift+F8**: Set mode "NULL to Unselected" (Texture Tool)
- **Shift+F9**: Set mode "Align to View" (Texture Tool)
- **Shift+Del**: Cut the selection and put it on the Clipboard
- **Shift+Del**: Delete current path
- **Shift+Ins**: Insert Clipboard contents
- **Shift+Tab**: Decrement 2D layout / 3D rendermode
- **Ctrl+Alt+S**: Save the active document with a new name containing incremented version number
- **Ctrl+Alt+T**: Triangulate faces (Vertex Manipulation)
- **Ctrl+Alt+W**: Weld vertices (Vertex Manipulation)
- **Ctrl+Shift+A**: Align texture on selected faces to 3D camera's view
- **Ctrl+Shift+B**: Snap each selected object to the grid individually (without altering size/relativity)
- **Ctrl+Shift+Backtab**: Switch to the previous document window
- **Ctrl+Shift+C**: Carve using the selected objects
- **Ctrl+Shift+D**: Make selected brushes detail (set CONTENTS_DETAIL flag)
- **Ctrl+Shift+E**: Convert current path to entities
- **Ctrl+Shift+F6**: Switch to the previous document window
- **Ctrl+Shift+G**: Selects a brush by number
- **Ctrl+Shift+H**: Make each of the selected solids a group of hollow objects
- **Ctrl+Shift+I**: Invert matrix of the selected curve brushes
- **Ctrl+Shift+L**: Toggle texture locking for scaling

- **Ctrl+Shift+M**: Merge the selected objects
- **Ctrl+Shift+P**: Export contents on message window to file
- **Ctrl+Shift+Q**: Toggle cull tree visibility
- **Ctrl+Shift+R**: Rotate the selected objects clockwise
- **Ctrl+Shift+S**: Make selected brushes structural (clear CONTENTS_DETAIL flag)
- **Ctrl+Shift+T**: Transpose matrix of the selected curve brushes
- **Ctrl+Shift+U**: Toggle cull tree
- **Ctrl+Shift+W**: Toggle group ignore
- **Ctrl+Shift+Z**: Redo the previously undone action
- **F**: Toggle texture filtering
- **H**: Hide the selected objects temporarily
- **O**: Toggle clipping sizes (Clipping Tool)
- **P**: Toggle 3D-grid
- **Q**: Toggle depth test for selection overlay
- **U**: Show all visgroups
- **X**: Toggle 3D auto-selection by mouse dragging
- **Z**: Toggle free-look mode
- **F1**: Open reference manual (this file)
- **F2**: Configuration options
- **F3**: Edit effect/shader scripts
- **F4**: Export level again to the same format
- **F5**: Synchronize local files with the remote storage
- **F6**: Object Bar
- **F7**: Texture Bar
- **F8**: Filter Control
- **F9**: Run the map in the game
- **Back**: Switch to the previous document window
- **Break**: Interrupt compile process (Compile Process Window)
- **Tab**: Increment 2D layout / 3D rendermode
- **Del**: Delete marked objects
- **Del**: Delete selected path nodes
- **Forward**: Switch to the next document window
- **Ins**: Add path nodes before selected nodes
- **PgDown**: Next camera (Camera Tool)

- **PgUp**: Previous camera (Camera Tool)
- **[**: Lower the size of the grid
- **]**: Raise the size of the grid

6. F.A.Q.

This section will grow appropriately along with questions we receive from you. Feedback address is: support@hlfx.ru. You can also ask your questions at Steam Community: <http://steamcommunity.com/app/496450>; or at our Internet forums (in Russian): <http://hlfx.ru/forum>.

6.1. General Questions

Quake II/III textures are not displayed — what's wrong?

You have to unpack game archives (pak-files) using one of popular programs, e.g. PakExplorer. Quake III archives (pk3-files) can be unpacked with a popular archiver WinZip or WinRAR. Unfortunately the current version is not capable of loading packed textures, although it can load packed models and sprites.

Why do I experience strange errors importing MAP format?

MAP is the only format most compilers are familiar with. Despite its standard suffix, the format slightly differs between different games. Although plugins do their best to detect format version and read only the necessary data, sometimes they fail. That is why you should import MAP files using the game configuration they were saved in. Besides, the formats of games not supported at this moment could also be read erroneously. Please note that sometimes there will be no errors on importing the MAP using the incompatible configuration, but some information will be lost. For example, this occurs on importing a Quake II MAP as a Half-Life level.

Where are configuration files located?

Configuration files are usually located in the directory you installed J.A.C.K. into (under “SteamApps” directory for the Steam version). However if the editor was installed into the system directory (e.g. “Program Files”), it can lack for write access rights. In this case the configuration files are located in the “My Documents/JACK” directory (or, “/home/username/Documents/JACK” under Linux).

Sound playback in Linux doesn't work. What's wrong?

Editor uses NAS (Network Audio System) for sound playback. You have to install necessary packages (for example, “#apt-get install nas nas-bin” in Debian/Ubuntu) and enable OSS emulation (execute “\$modprobe snd_pcm_oss”). You'd better add snd_pcm_oss module to automatically loaded ones, so you won't have to run the command each time you reboot. After these operations sound should magically appear.

6.2. Half-Life Questions

Which format is better to store levels in — JMF or RMF?

Despite the similarity of suffixes, these formats are fundamentally different. JMF is a native format of the editor; all the information that is to be saved, is stored into the JMF, and no plugins are required to deal with it. RMF is a proprietary rich map format designed for Worldcraft/Valve Hammer Editor; since its structure is known, J.A.C.K. supports it within a Half-Life plugin. Both formats are available for import and export, however it is recommended to use JMF to store your map sources. RMF format doesn't support some peculiar J.A.C.K. features, e.g. multiple VisGroups per object. Use the RMF format to transfer your map from J.A.C.K. to VHE. Besides the RMF can be utilized as a backup format.

6.3. Quake/Hexen II Questions

There are no textures, where can I get them?

You have to extract the textures into a separate WAD file. Please use a corresponding command from the Plugins → Quake menu.

Models are not animated. Is that intent?

In Quake, animation frames are switched manually in the game logic code. You can select any frame for editing purposes by altering a “sequence” keyvalue. Animation groups (e.g. flame) and rotating items should be animated correctly.

I get an error “Invalid argument” during a compile process. What's wrong?

At present the compilers distributed with the editor do not support multiple WAD files. To solve the problem, either place all textures into a single WAD file, or use third-party MAP-220 compatible compilers.

6.4. Quake II Questions

Models are not animated. Is that intent?

In Quake II, animation frames are switched manually in the game logic code. You can select any frame for editing purposes by altering a “sequence” keyvalue.

6.5. Quake III Questions

How to create a curve?

Use a Block Tool, but select “(curves)” in the Categories section, and then select a curve type in the Objects menu.

What is the difference between q3map and q3map2?

Q3map compiler is an original map compiler for Quake III: Arena, it is present in the package only for compatibility reasons. It is recommended to use an improved, community-developed q3map2 compiler, which is also present in the packet. Please note that both compilers were modified to maintain editor's export format compatibility.

Why do some shaders issue loading warnings?

The editor supports a standard Quake III shader format, but some shaders were developed for earlier versions of the engine. You may ignore there warnings as the engine does. The warning "no appropriate stages for vertex coloring mode" means that the editor failed to select a shader stage to apply selection color to. Such surfaces will not be highlighted in a red color when selected in shader animation mode.

7. External References

7.1. Official Web Pages

- <http://jack.hlfx.ru/> — official J.A.C.K. website.
- <http://crystice.com/jack/> — J.A.C.K. on the Crystice Softworks website.
- <http://store.steampowered.com/app/496450> — Steam Store page; here you can purchase the Steam version of J.A.C.K. and get all the benefits of frequent automatic updates, SteamCloud storage, auto-configuration tool and more.
- <http://steamcommunity.com/sharedfiles/filedetails/?id=517677347> — Steam Greenlight page.
- <http://hlfx.ru/forum/forumdisplay.php?s=&forumid=35> — official Support Forums (in Russian; although posts in English are also welcome).

7.2. Map Compilers

- <http://forums.svencoop.com/forumdisplay.php/217-Vluzacn-s-Map-Compile-Tools> — latest Half-Life map compiling tools, fully compatible with J.A.C.K.

7.3. Additional Game/Engine/Mod Support

- <https://bitbucket.org/FreeSlave/def2fgd> — def2fgd; the tool for converting “.def” and “.ent” files used by GtkRadiant/NetRadiant to “.fgd” files used by J.A.C.K. editor.
- <https://sites.google.com/site/svenmanor/> — Sven Co-op Half-Life modification level design manor; mapping tutorials and resources using J.A.C.K. editor.
- http://q3map2.robotrenegade.com/docs/shader_manual/ — Quake III Arena shader manual, with Q3Map2-specific keywords.
- http://jack.hlfx.ru/en/articles/4/jack_trem.html — a bit obsolete, but still informative lesson about how to configure and use J.A.C.K. for the game Tremulous.