Advanced Compiling :: [Jv\_map](http://www.mohaaaa.co.uk/phpBB2/profile.php?mode=viewprofile&u=146)  
-------------------------------------------------------------------------------  
  
Every map you create with [MOHRadiant](http://personalserver.fileplanet.com/redir.asp?from=PMOH&to=http%3A//www.fileplanet.com/index.asp?file=86998) or any other editor needs to be compiled first before you can play it in game. This tutorial descibes two ways of compiling a map, with [MBuilder](http://www.fileplanet.com/dl/dl.asp?/planetmedalofhonor/map/mbuilder.zip) and directly from a command line. There's also some basic info about how the compiler works and the tutorial covers all available options you can choose from when compiling.

The compiling process consists of three separate stages, known as bsp, vis and light. After the bsp stage your map should be ready to run in the game. The vis and light stage just improve the end result. Actually, the bsp stage turns the text based .map file into a binary space partition (.bsp), a file format that can be read by the game. In the bsp phase, the map is also split up into vis leaf nodes, these are saved into a .prt file. This means that a map with a complex architecture of structural brushes will take more time in the .bsp phase than a well-detailed map. The vis stage calculates the visibility table, which is saved into a .vis file. With this visibility table, the game is able to determine what triangles are visible from what viewpoint. The faces that can't be seen won't be drawn, so the vis phase should mean a drastic improve to the overall framerate when playing the map. The light stage generates lightmaps for every surface, so you will see different shades on different surfaces. When you want to light with radiosity, you need to run a vis compile first. When your compile is done, you may remove the .prt and .vis files. There's no reason to distribute them along with your map.

Probably the easiest way to compile a map for an inexperienced mapper (and still an easy way for an experienced mapper ofcourse) is to compile with a front-end tool like [MBuilder](http://www.fileplanet.com/dl/dl.asp?/planetmedalofhonor/map/mbuilder.zip). This program has a user interface that's very easy to use. Nevertheless, working with [MBuilder](http://www.fileplanet.com/dl/dl.asp?/planetmedalofhonor/map/mbuilder.zip) is explained in the [Basic Compiling](http://www.mohaaaa.co.uk/mohaa/tutorials/basic_compile.php) tutorial. Another way of compiling is compiling directly from a command line, for example a DOS box. Easier is to create a batch file that does the job for you, that's described below as well..

-------------------------------------------------------------------------------  
Compiling from the command line  
-------------------------------------------------------------------------------

Compiling from a command line is slightly more difficult, but allows more freedom when compiling. To completely compile your map, you need a command for the bsp, vis and light stage. Both the bsp and vis stage are processed by q3map.exe, the light stage is executed by a different program called mohlight.exe. The basic syntaxis for all three commands is the following:

*q3map -gamedir "[MohaaDir]" "[MohaaDir]/main/maps/mymap.map"  
q3map -vis -gamedir "[MohaaDir]" [MohaaDir]/main/maps/mymap"  
mohlight -gamedir "[MohaaDir]" "[MohaaDir]/main/maps/mymap"*

Where [MohaaDir] should be replaced with the path mohaa.exe is in, for example

*q3map -gamedir "E:/Games/Mohaa" "E:/Games/Mohaa/main/maps/mymap.map"*

You can add any special options before the map's filename. For example:

*q3map -gamedir "[MohaaDir]" -blocksize 0 "[MohaaDir]/main/maps/mymap.map"*

Will compile with -blocksize set to 0. See the [Compiling Options](http://www.mohaaaa.co.uk/mohaa/tutorials/compile.html#compopt) section for more information about the different compiling options.

When you don't want to type all these commands over and over again, it's a good idea to make a batch file. Just copy all commands (with full path info) into a text editor and save it as a .bat file. Then double click that .bat file to compile the map.

-------------------------------------------------------------------------------  
Compiling Options  
-------------------------------------------------------------------------------

(A) BSP Stage  
(i) [Common Options](http://www.mohaaaa.co.uk/mohaa/tutorials/compile.html#a_1)  
(ii) [Additional Options](http://www.mohaaaa.co.uk/mohaa/tutorials/compile.html#a_2)

(B) VIS Stage  
(i) [Common Options](http://www.mohaaaa.co.uk/mohaa/tutorials/compile.html#b_1)  
(ii) [Additional Options](http://www.mohaaaa.co.uk/mohaa/tutorials/compile.html#b_2)

(C) Light Stage  
(i) [Common Options](http://www.mohaaaa.co.uk/mohaa/tutorials/compile.html#c_1)  
(ii) [Additional Options](http://www.mohaaaa.co.uk/mohaa/tutorials/compile.html#c_2)

-------------------------------------------------------------------------------  
(A) BSP Stage  
-------------------------------------------------------------------------------

(i) Common Options

**-gamedir [gamedir]**: Make this point to the directory Mohaa.exe is in. You may as well use relative paths (e.g. ../..).  
**-blocksize [blocksize]**: Sets the size of block portals. These are automatically generated portals and are usually not beneficial for the -vis compile speed nor for the frame rates when playing your map. This setting defaults to 1024 but can be set to 0 to disable blocksize portal splits, which I would strongly recommend.  
**-onlyents**: only updates the entities in your map without changing the architecture. Very useful when you only changed some entities and don't want to wait a long time for your compile. Doesn't work with lights though.  
**-nohint**: all hint brushes are ignored, no additional portal splits are created.  
**-snapdistance [distance]**: This option snaps vertices to the grid that are within [distance] units away from the grid. Defaults to 0.1 units.  
**-nostatic**: Disables the addition of Static Collision Masks to the scene. These are .map files in the same directory as static tikis in pak0.pk3. These masks consist of different material type clip brushes which tell the game what kind of debris has to be spawned when the player shoots at them. Compiling with the -nostatic option will cause all static models to go non-solid.  
**-visiblestatic**: Makes the Static Collision Masks use a visible shader. This is useful for when you're creating your own collision masks. When testing your map, you might have to type 'r\_drawstaticmodelpoly 0' in the console to see the masks.

(ii) Additional Options

**-v[erbose]**: increase the text output when compiling to get additional information on what's currently happening, but slows down the compile.  
**-threads**: Sets the number of CPU threads to use. If not specified, all CPUs in the computer are used. I'm not sure if this makes any difference, because I only have one processor.  
**-fulldetail**: all detail brushes are considered structural. Never use this option, unless you'd want to wait hours for your vis-compile.  
**-nodetail**: detail brushes are ignored, so you'll get a result that looks like your map in Radiant with the detail filter on.  
**-detailterrainborders**: force terrain mesh boundaries to be flagged max detail.  
**-notjunc**: disables automatic t-junction fixing. T-junctions are very small 'sparkles' that you can sometimes see when brushes don't exactly fit. If you run q3map with -notjunc, you have an increased chance of seeing these sparkles.  
**-moddir [moddir]**: mod subdirectory of gamedir.  
**-chopblocklast**: do blocksize chops after other chops.  
**-nowater**: water brushes (top or bottom texture has surfaceparm water) are ignored.  
**-nocurves**: all curves and patches are ignored, they will not show up in game so to say.  
**-nosubdivide**: ?  
**-nomerge**: ?  
**-expand**: ?  
**-tmpout**: ?  
**-fakemap**: ?  
**-all**: ?  
*Note: If you know what any of the options marked with a '?' do, please drop me a line.*

-------------------------------------------------------------------------------  
(B) VIS Stage  
-------------------------------------------------------------------------------

(i) Common Options

**-vis**: just don't forget this one, otherwise you're running a BSP process.  
**-gamedir [gamedir]**: Make this point to the directory Mohaa.exe is in. You may as well use relative paths (e.g. ../..).  
**-nomanvis**: Disable manual vis, such as vis leaf groups. I must admit I don't know much about manual vis, because I always use automatic vis, but if you have anything to add just tell. **(This option is not tested)**

(ii) Additional Options

**-threads**: Sets the number of CPU threads to use. If not specified, all CPUs in the computer are used.  
**-v[erbose]**: increases text output, slows down the compile.  
**-fast**: all view portals are visible, very fast but may have severe performance impact.  
**-moddir [moddir]**: mod subdirectory of gamedir.

-------------------------------------------------------------------------------  
(C) LIGHT Stage  
-------------------------------------------------------------------------------

(i) Common Options

**-gamedir [gamedir]**: Point this to the directory Mohaa.exe is in. You may as well use relative paths (e.g. ../..).  
**-bounce [num]**: Sets the number of radiosity bounces to do. Defaults to 8. Radiosity means that every texture emits light if it's lighted, for example, a red curtain will color the surrounding walls slightly red. This light red will bounce off the wall to other brushes, until the number of bounces is reached.  
**-final**: Sets all lighting options to full detail except -chop.  
**-fast**: Loses some lighting quality for a faster compile.  
**-staticshadows**: Makes static models cast shadows. This can make your map look a lot more realistic, but will add unbelievably to the compile time. With -staticshadows, you can expect your compile to take about 4 times as long.

(ii) Additional Options

**-moddir [moddir]**: mod subdirectory of gamedir.  
**-threads**: Sets the number of CPU threads to use. If not specified, all CPUs in the computer are used.  
**-v[erbose]**: Enables verbose message output (you get more info)  
**-area [num]**: Sets the scale of surface lights. Default is 1.0 Surface lights are all light emitting textures.  
**-point [num]**: Sets the scale of point lights. Default is 1.0 Entity lights are also known as point lights.  
**-notrace**: Don't do any occlusion testing at all (no shadows).  
**-nosurfshadows**: Brush surfaces don't cast shadows.  
**-nocurveshadows**: Curved surfaces don't cast shadows.  
**-nopatchshadows**: Terrain patches don't cast shadows.  
**-nocurvelight**: Don't light curved surfaces.  
**-nopatchlight**: Don't light terrain patches.  
**-nodiffusesun**: Don't do the diffuse sunlight calculations. Sun diffuse can be set in the worldspawn.  
**-nogrid**: Don't calculate the entity lighting grid. The lighting grid is used to adjust the brightness of entities, even when they move around, such as the player's gun getting darker in shadowed areas.  
**-radgrid**: Calculate radiosity into the Light Grid.  
**-vertex**: Enable vertex lighting of the map.  
**-randomdir [dir]**: Sets the amount of random lighting direction. Default is 0.  
**-chop**: Sets the amount to chop up radiosity patches. Default value is 64.  
**-minchop**: Sets the smallest size to chop a radiosity patch to. Default value is 64.  
**-smoothangle**: Sets the maximum angle to smooth across. Default is 45.  
**-extra**: Does high detail light sampling. Only increases sunlighting quality, so don't use this, but use final instead.  
**-extragrid**: Does a full detail calculation of the entity lighting grid.  
**-border**: Add debugging borders to the lightmaps.  
**-raddebug [num]**: Different debug options when in range 1-4. Never tried any of these, but they may be useful.  
**-radscale [num]**: Radiosity scale factor. Default is 0.95, 0 is no radiosity. Don't set it too high, it will cause a 'fun fair' look.  
**-attenuate [num]**: Sets atmospheric attenuation over distance in percent. Attenuation A means radiosity reduces by A% every 16 units, in addition to the 1 / distance squared falloff. Default is 4. If you have a lot of fog in your map, set this value higher, in a clear atmosphere, set it a bit lower.  
**-onlygrid**: Only calculates the lighting grid.  
**-ambient [RGB]**: Sets the RGB ambient light on a 0 - 255 scale. Crashes Mohlight on my system, but you can set it in the worldspawn with 'ambientlight'.

Need any help : Ask in the [Forum](http://www.mohaaaa.co.uk/mohaa/forum/index.php)