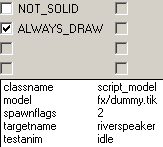
Adding Custom Sounds :: [jv\_map](http://www.mohaaaa.co.uk/mohaa/forum/profile.php?mode=viewprofile&u=146)  
-------------------------------------------------------------------------------

This tutorial describes how to add custom sounds to a map, such as positional ambiences like radio, tree or river sounds. This can be done entirely through scripting but in most cases you'll rather want to place some kind of sound entity in Radiant and refer to this object in the script.

First I'll explain certain general concepts of sounds in mohaa. Any entity can make a sound (even multiple sounds simultaneously if you want) provided it has a model. This means a script\_origin can't emit a sound nor can any of the speaker entities in the Radiant right-click menu. It's common practice to use a script\_model with model 'fx/dummy.tik' as invisible speaker entity.

Now let's get started with the actual tutorial. I'll give a step-by-step description on how to make the custom sound 'river.wav' in the 'main/sound/amb/' folder play in a demo map.

First of all add a script\_model where you want the sound to play loudest. Press 'N' to bring up the entity inspector and add key 'model' / value 'fx/dummy.tik'. Since you want to control the entity from script it's necessary to add a targetname. For instance add key 'targetname' / value 'riverspeaker'. As the Quake 3 engine treats your new sound entity like any other entity it will be hidden when it's not inside your 'pvs'. In short this means you may not hear the sound when the sound entity is not in sight. Check the 'ALWAYS\_DRAW' checkbox so this won't happen.



Well that was easy wasn't it? Now on to the scripting part which is slightly more complicated, so you can easily compile your map in the meantime. If you don't have a script for your map yet, just make a new file called 'mymapname.scr' in the same folder as the bsp file of your map. You can edit a script in any text editor.

There are basically two things that need to be scripted for the sound to work. First, your sound has to get an 'alias', which is a (shorter) name for the sound that is used in the script instead of the filename. This alias also contains information about volume, pitch, audible radius and possibly a subtitle for your sound. Once the alias is defined you can use it in your script. This is also the second thing: getting your sound to play.

First we'll do the alias. The basic syntaxis for this is the following:

local.master = spawn ScriptMaster  
  
local.master aliascache [aliasname] [soundfile] soundparms [basevolume] [randvolume] [basepitch] [randpitch] [mindist] [maxdist] [channel] loaded maps "obj dm"  
  
// add any number of additional aliases here

The values for all soundparms (basevolume, randvolume etc.) are discussed in the stock file ubersound/ubersound.scr in pak0.pk3 (you can open this file with e.g. winzip). Within the scope of this tutorial only [aliasname] and [soundfile] are interesting, and they're usage is very straightforward. [aliasname] is the name of the alias which we will use in the script. For example make it 'snd\_river'. The [soundfile] is simply the real path to the sound and the filename, in this example it should be sound/amb/river.wav. The final line could look like this:

local.master aliascache snd\_river sound/amb/river.wav soundparms 1.0 0.2 1.0 0.2 200 2000 auto loaded maps "obj dm"

Your script should now look similar to this:

main:  
  
  local.master = spawn ScriptMaster  
  
  local.master aliascache snd\_river sound/amb/river.wav soundparms 1.0 0.2 1.0 0.2 200 2000 auto loaded maps "obj dm"  
  
  level waittill spawn  
  
end

**Note:** If you use this way to define aliases (instead of in ubersound.scr or uberdialog.scr) you can only use 'aliascache', just 'alias' won't work. Also if you want a sound to play on the local channel (no 3D localization) you'll have to write "local" (between quotes).

Now it's time to actually start the sound. In most cases you'll want a continuously looping sound. Scripting is very easy and requires not more than one line:

$riverspeaker loopsound snd\_river

'$riverspeaker' points to the entity you made in Radiant with this targetname.

If you want more advanced sound effects than the boring looping sound, you can use the playsound, stoploopsound or stopaliaschannel commands. With some practice these commands are fairly easy to understand.

This is what your 'final' script could look like:

main:  
  
  local.master = spawn ScriptMaster  
  
  local.master aliascache snd\_river sound/amb/river.wav soundparms 1.0 0.2 1.0 0.2 200 2000 auto loaded maps "obj dm"  
  
  level waittill spawn  
  
  $riverspeaker loopsound snd\_river  
  
end

At any time the aliases have to be defined above 'level waittill spawn' (except if the alias refers to a pre-defined sound but with different soundparms) whereas the loopsound command has to be placed below the 'level waittill spawn' line.

Enjoy your new sound!

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