

MiriX Tutorial

Manual rev. 0.3, 18-02-'04

Authors: [Rob "RKone" Morriss](#)
[Skwasjer](#)

For MiriX v2.0

1. Introduction.....	2
2. Before you start	3
3. MiriX Overview	4
4. Building maps for MiriX	5
4.1. Start at the center	5
4.2. Mapping with separate .map files	5
4.3. Creating entities for MiriX	5
4.3.1. Angles	5
4.3.2. Base-specific keys	6
4.3.3. Lights	7
4.3.4. Target- and targetname-keys	8
4.3.5. Team Arena entities.....	8
5. Using MiriX	9
5.1. Team options.....	9
5.1.1. Swap textures/shaders	10
5.1.2. Swap models	10
5.1.3. Adjust team items	10
5.1.4. Adjust lighting (colors)	10
5.1.5. Team item spawnflags.....	10
5.1.6. 3w/Alliance spawnflags.....	10
5.1.7. Adjust Red/Blue message	10
5.1.8. Honor "*_mirix"-key/value	10
5.2. Rotation or flip	11
5.3. Substitution lists.....	11
5.4. Keys on the worldspawn-entity.....	12
5.5. Building the final map	12
6. About this document.....	13

1. Introduction

Welcome to both this manual and tutorial for MiriX. It's been more than two years now that this tool has been out in the world, but it has been without a decent manual. To my knowledge only few mappers know of this tool, and even less actually have tried it. Strange if you understand what this tool can do for you.

Somewhere in July of 2002 Rob "RKone" Morriss who was trying MiriX on his fairly large CTF-map approached me. Rob at first struggled with MiriX due to the lack of a manual describing what should be done for the best end result. His experience as a mapper and mine as a developer have now led to this manual. I say great thanks to him for his work. If you want to see his work, visit http://rkone_maps.tripod.com. In case you don't still know who he is, he is the author of X3Map06: The Sound of Inevitability (<http://x3maps.gameplayer.org>), and it was this map he used MiriX with.

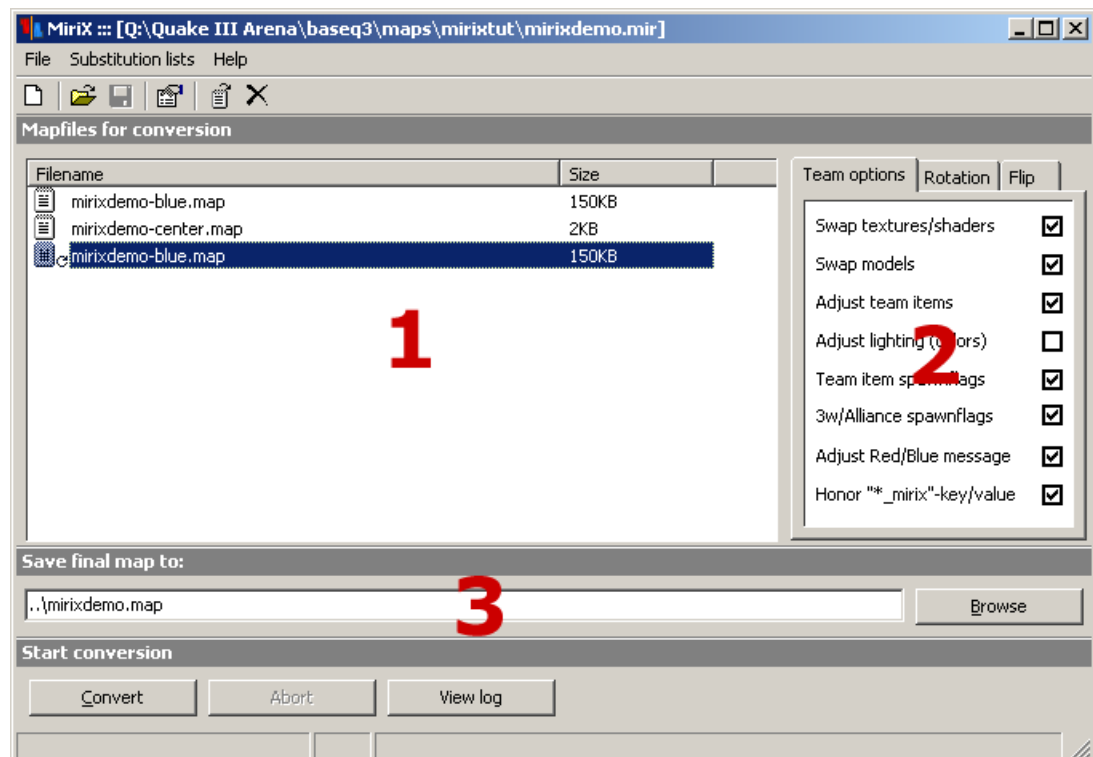
2. **Before you start**

First off, we assume you have made maps in the past and have basic/moderate mapping skills.

This tutorial assumes you will be creating the blue base (so we refer to “the blue base” from here), and using MiriX to make the red base. MiriX works just as well the other way around (iow: creating the red base and using MiriX to copy it to a blue base). The tutorial comes with a demo-map, which you can use as a reference. (Copyright notice for the map Rob???)

3. MiriX Overview

The main thing to understand about MiriX is that it is a utility that simply joins map-files (.map extension, Q3/GTKRadiant mapformat) together. The real power behind MiriX is its ability to automatically change *all* a map's CTF textures and entities with one simple mouse click. With MiriX, you can spend hours making changes to the blue base, and then create a complete map with a perfectly rotated (or mirrored) red base - **in seconds**. The rotation or flip options are customizable and even allow you to make L-shaped maps, or others sorts of combinations of rotation and flip, while keeping textures and entities in place. Let's look at the MiriX interface:



The three main sections are highlighted in red.

- 1 Map file List - The list of .map files to be joined together
- 2 Modifiers - Operations to be applied on the selected map file
- 3 Output - Output map file name

In the example we're using MiriX to create a complete CTF map using only a blue base and a center area. In the Map file list we have:

- mirixdemo-blue.map - the blue side of the final map
- mirixdemo-center.map - the center portion where the two sides join
- mirixdemo-blue.map (second instance) - this is the same file as the first instance, but has Modifiers applied to it, to convert it to the red base. All options will be explained in the next chapter.

The Output textbox points to where the final combined map is stored.

4. Building maps for MiriX

First of all, keep in mind that brushprimitives are currently not supported. Support is not included because it still seems to be an experimental feature in GTKRadiant. Even though it might be stable, I decided not to put any effort into it yet. Same goes for the new xml map format (.xmap), which is a new format still being worked on (to my knowledge).

Now lets build a map...

4.1. Start at the center

All map files must share a common origin. MiriX will support any point as an origin, but it's easiest to turn on the grid numbering in Radiant and orient the map so the center is at 0, 0, 0.

4.2. Mapping with separate .map files

The most difficult task when building your map using MiriX is creating one map in separate files. Radiant only displays one map at once, and it's hard enough to create a map without leaks when you can see all the brushes! One method is to build the blue base and center areas together, and then later cut & paste the center area to a new map file. This is the best method to use early on. However as the map gets more complicated, it becomes more difficult to determine which brushes belong to the center, and which ones belong to the blue base.

Tip: in early design stages, enclose your base in one huge box made up of structural caulk brushes, and leave the side out to which the other base is duplicated. This will avoid early leaks, and eases your early development. However, it may be hard to repair leaks in the end. Up to you to decide.

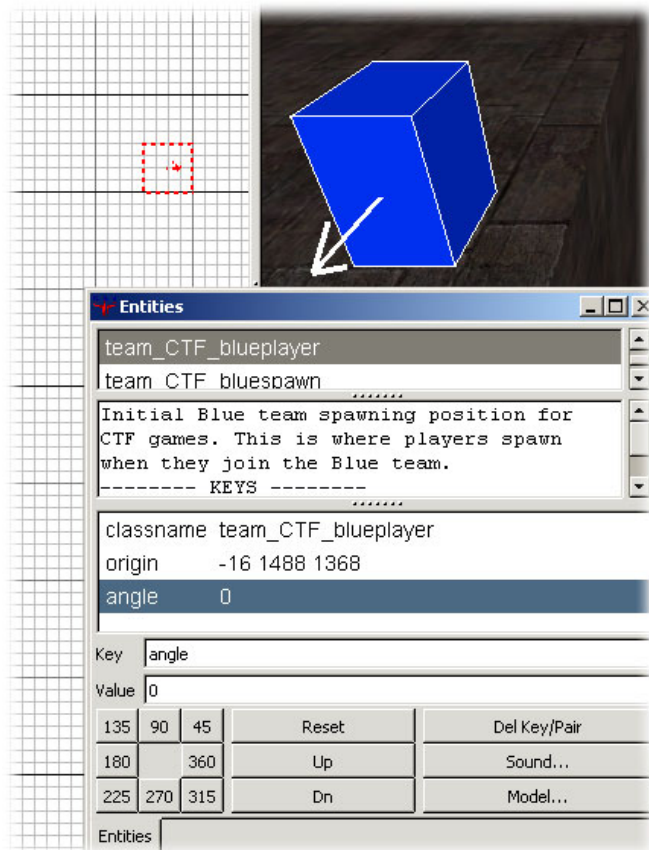
4.3. Creating entities for MiriX

For MiriX to be able to copy a blue base to a red base, certain preparations must be taken care of by the mapper. There are several entities that need to be changed when they are duplicated to for instance avoid collision on target(name)s, entities that should point to a different direction (angle-key), etc. This section describes all supported features and the required preparation.

4.3.1. Angles

A lot of entities have the so-called **angle**-key, for instance misc_models and team_CTF_bluespawn. This key is used to point an entity to a certain direction. Whenever a base is duplicated and rotated/flipped, the entity should not only be repositioned on the grid, its orientation should also change. MiriX can recalculate the new angle for you.

Note: entities may have the proper orientation without an **angle**-key defined, but in such a case MiriX will not recalculate the new angle when creating the red base. If the entity is properly oriented without an **angle**-key, then give the entity an **angle**-key with value "0".



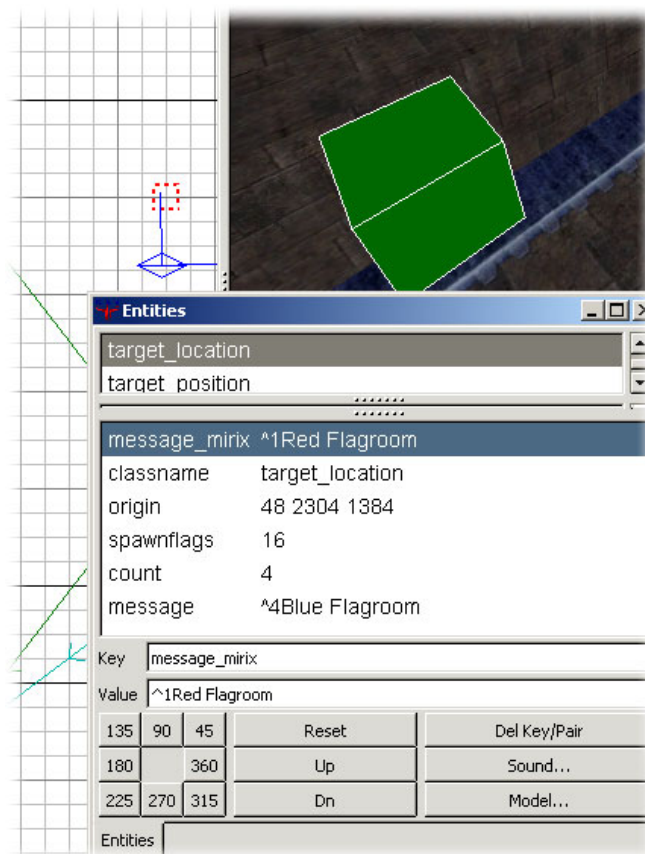
If it's got an arrow, it needs to have an angle key. Don't forget models!

4.3.2. Base-specific keys

While creating the blue base, you must keep in mind that you're also building the red base. Many entities have keys that are specific to one side, for example a `target_location` entity with the **message**-key "Blue Flagroom". Of course we want this changed for the red base to "Red Flagroom".

To change the values for suchs keys MiriX has the special ***_mirix** key. Create a new key with the same name, but add `_mirix` to the key name, and then assign the value as you were creating the key for the red base. For our `target_location` example, there will be a **message**-key with the value "Blue Flagroom", and another key, **message_mirix** with the value "Red Flagroom".

Note: the **message**-key can also be automatically translated by enabling the Adjust Red/Blue message-option (see 5.1.7).



Notice I've also added **count_mirix** for use with the Threewave mod's **count** key. Obsolete: MiriX does this automatically.

Important keys to remember for using the *_mirix key

Entity	Key for blue base	Key for red base	Description
target_location	message	message_mirix	Change base specific location information
light	_color	_color_mirix	To change blue tinge to red tinge
func_door	team	team_mirix	To avoid having 2 pairs of doors being teamed
target_print	message	message_mirix	Doubt it's used a lot in CTF-modes, but still something to remember
Any more???			

4.3.3. Lights

MiriX can automatically change the RGB values of all the light entities in a map-file, so they are ordered BGR in the red base, effectively turning lights with a blue tinge into lights with a red tinge. The drawback is that you cannot specify which lights get changed, you can only choose all or none. This would cause problems, for example, if you have yellow lights to brighten a section of the blue base that your skybox's sun isn't reaching. It is therefore strongly recommend that you use _color_mirix to manually specify which lights have their colors changed (see Base-specific keys for more info).

4.3.4. Target- and targetname-keys

MiriX will automatically recreate the target -> targetname relations, but only if the default targetname naming convention (t1, t2, ..., t582, ...) is used. If you manually specify a **targetname** value (like "myjumppad"), MiriX will not change it, resulting in the cross-linking of the entities (as the targetname key in the newly created red base will have the same value as the original in the blue base). This might be something you actually want in a certain situation but that's up to the mapper. We advice to let Radiant create all your target/targetname links automatically using the **Ctrl-K** hotkey.

4.3.5. Team Arena entities

MiriX recognizes the spawnflags key used by TA entities:

- item_doubler
- item_ammoregen
- item_guard
- item_scout

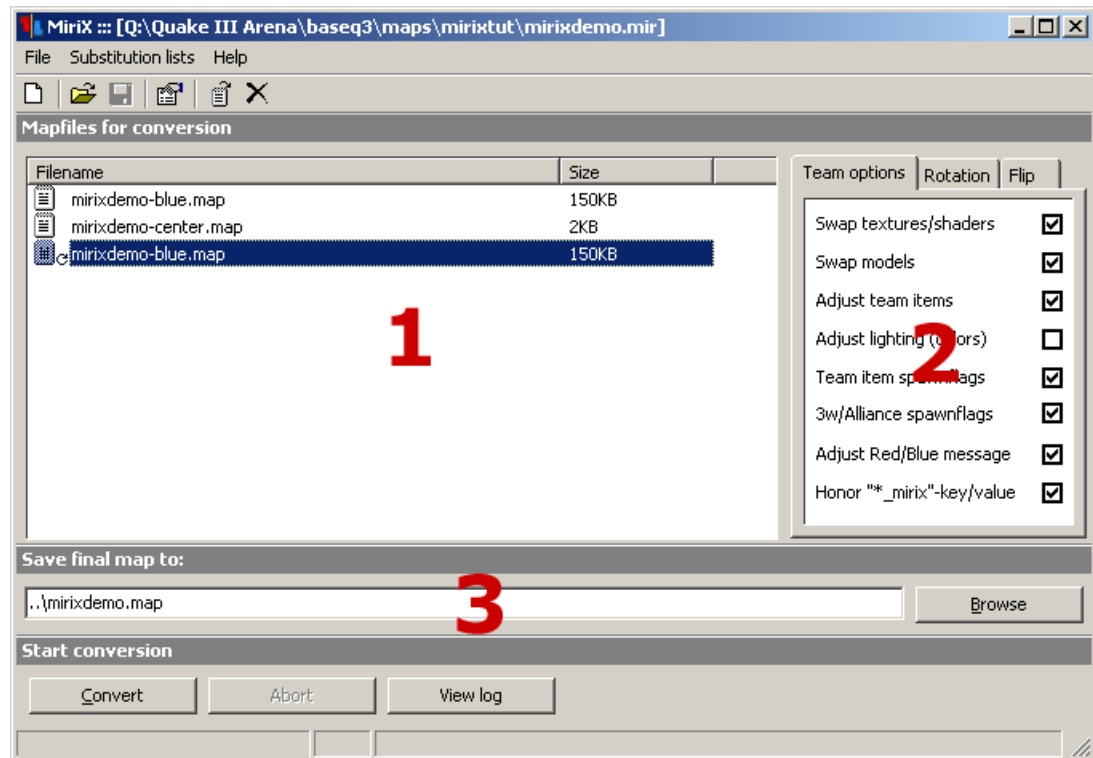
Note: These entities do not require a **spawnflags_mirix** key.

If you specified that, for instance the doubler-entity in the blue base is for the blue team only, the spawnflags on the copied instance for the red base will be changed to only allow the red team to pickup the doubler.

5. Using MiriX

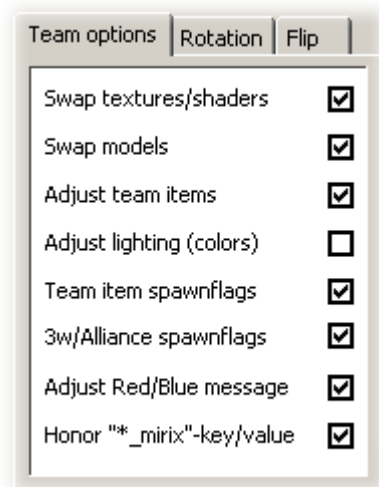
The first step to use MiriX is to tell it which map-files you want to use. Use the Add Map button (second last button, the one that looks like notepad with an arrow above it) to add a mapfile to the list. The order the files are entered do not matter, however with one exception, see Keys on the worldspawn-entity.

Add a 2nd copy of the blue base - this is where MiriX will get its information on how to build the red base.



5.1. Team options

Now select the 2nd copy of the blue base, and choose the modifiers you want MiriX to use:



5.1.1. Swap textures / shaders

Change a texture/shader that matches an entry in the substitution lists to their counterparts also specified in the substitution list. See Substitution lists for more info.

5.1.2. Swap models

Change a model that matches an entry in the substitution lists to their counterparts also specified in the substitution list. This applies to any entity with a **model**-key or a **model2**-key. See Substitution lists for more info.

5.1.3. Adjust team items

With this option enabled MiriX will change the following entities vice versa:

team_CTF_blueflag	team_CTF_redflag
team_CTF_blueplayer	team_CTF_redplayer
team_CTF_bluespawn	team_CTF_redspawn
team_blueobelisk	team_redobelisk

5.1.4. Adjust lighting (colors)

Change `_color` values in light entity from RGB to BGR. For instance "0.8,0,0" to "0,0,0.8". The `_color` key on the worldspawn-entity will be ignored.

For more control over which light-entity gets changed, do not use this automatic option. Instead use the `*_mirix`-key method as described in Base-specific keys.

5.1.5. Team item spawnflags

MiriX can automatically change the spawnflags key on the following teamspecific entities:

target_print
item_doubler
item_guard
item_scout
item_ammoregen

5.1.6. 3w / Alliance spawnflags

Change the **count**-key and/or **spawnflags**-key on an `item_location`-entity, for Threewave-resp. Alliance-mod.

5.1.7. Adjust Red / Blue message

Automatically change the text in the message-key (for both `target_location` and `target_print`) if an occurrence of the text 'red' or 'blue' is found. Note that MiriX keeps the case intact, so 'Red' will become 'Blue', 'red' will become 'blue' and 'RED' will become 'BLUE' alternatively. Of course this works both ways.

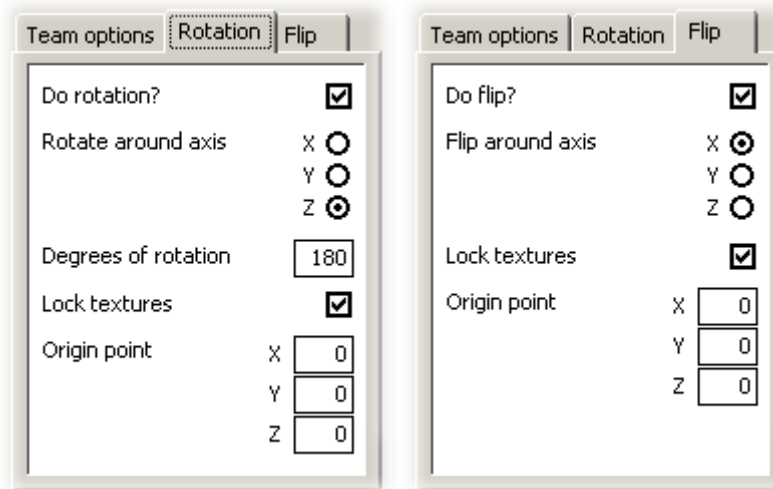
Added to that, MiriX will also change any occurrences of used inline color-coding. So ^1 will become ^4 and the other way around.

5.1.8. Honor "`*_mirix`"-key / value

If an entity has a `*_mirix` base-specific key, change it with the user-specified alternate value.

5.2. Rotation or flip

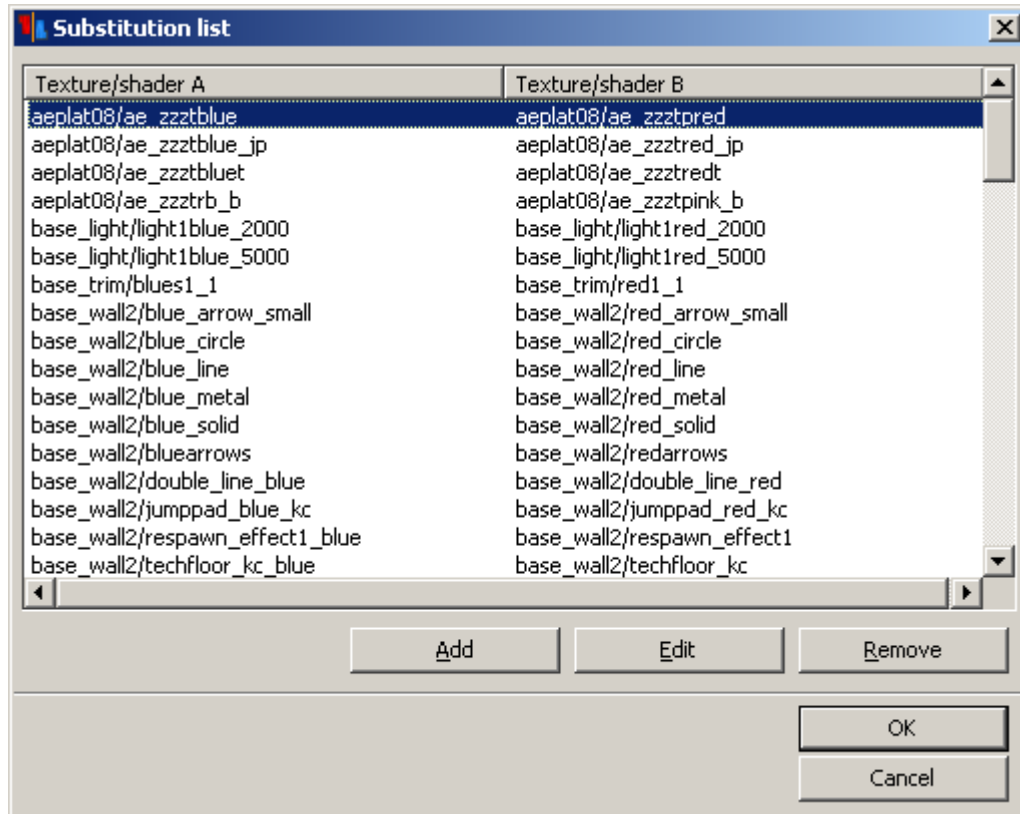
Next choose the Rotation tab if you want your red base to be a rotated copy of the blue, or the Flip tab if you're mirroring the blue base.



Check off only the one you wish to use. Rotations are normally about the Z-axis, and are 180 degrees. Flips are normally through the X- or Y-axis. If you have manually aligned textures, you'll want to lock textures so they stay in the same position after conversion. The Origin point defines the center of rotation for the map, in case you didn't build your map with 0, 0, 0 as the center.

5.3. Substitution lists

If your map uses custom textures/shaders and models that are team-specific, you can have MiriX change them automatically also. Add your items in the **Substitution lists** pull down menu. Alternatively, if you've got a texture that is being swapped by MiriX and you don't wish it to be (i.e. your stoplights are flashing blue in the blue base), find the texture/script or model in the substitution list and remove it.



texture/shader A is **blue**, texture/shader B is **red**.

5.4. Keys on the worldspawn-entity

Before we are ready to publish your final map you need to understand one more thing. As MiriX merges multiple mapfiles together, in effect it parses multiple worldspawn-entities and merges all the brushes and patches back to one. But how does it handle the keys on the worldspawn I hear you ask? MiriX will add all keys from the first mapfile it parses to the new worldspawn, and then add all keys from the next mapfiles. If duplicate keys are found, they will be omitted. Basically, just add all your required worldspawn-keys to the first mapfile in your mapfile-list and you're safe.

5.5. Building the final map

Finally, create a filename to save the output map to, and click Convert to start the process. It should take only a few seconds to complete. You may also want to save these MiriX settings for future use, with the **File/Save as** pull down menu option.

Done & ready for q3map! Now any future modifications to your blue base can be quickly mirrored to the red base by loading up your MiriX settings and clicking Convert again. Now if only there was a way to save this much time with q3map lighting... Added 18-02-'04, hah, wait, we got q3map2 now :-))))

6. About this document

- Quake III Arena, Team Arena, Q3Radiant are registered trademarks of Id Software, www.idsoftware.com
- GtkRadiant is a registered trademark of QERadiant, www.qeradiant.com
- If I forgot any credits: all productnames are registered trademarks of their respective owners.

This document and it's content are copyright protected.

© 2004 – Skwasjer. All rights reserved.

This document and it's content are provided as is, without any warranties. Distribution or reproduction is authorised, provided the source is acknowledged, save where otherwise stated.