

MAME ROMs Update

0. INTRODUCTION

1. TERMINOLOGY

2. SPLIT / MERGE. What does this mean?

- 2.1. Method 1: Split / Not Merged
- 2.2. Method 2: Split / Merged
- 2.3. Method 3: Not Split
- 2.4. Summary
- 2.5. Final Considerations

3. UPDATING ROMS

- 3.1. Why?
- 3.2. What do I need?
- 3.3. What is a fixfile?
- 3.4. What is a dat file?
- 3.5. Where can I get dat files?
- 3.6. What is a Rom Manager?
- 3.7. Where can I get Rom Managers?

4. ROM UPDATE USING CLRMAME PRO

- 4.1. Selecting the dat file
- 4.2. Loading the database
- 4.3. Configuring the ROM directories
- 4.4. Configuring the scanning options
- 4.5. Updating

5. ADDITIONAL INFORMATION

- 5.1. BIOS files
- 5.2. Fixfile creation: mamediff and the Rebuilder

6. ACKNOWLEDGEMENTS

0. INTRODUCTION

This guide has been written to help solve common problems that arise when installing a new MAME release and we find out that some games stop working correctly.

The information contained here has been verified using Windows 98, Windows 98SE and Windows 2000. There may be differences when using other operating systems.

Any questions, suggestions, corrections or complaints should be directed to the author at quemame@mixmail.com

1. TERMINOLOGY

Before we start, let's explain some terms we are going to use through this tutorial we must clearly keep in mind.

Game: just what it says!

Set: each of the versions of a game.

Clone: version of a game different from the original.

Example: The game “1941 - Counter Attack” has 2 *sets*, the *main set*: “1941 - Counter Attack (World)” and a *clone set*: “1941 - Counter Attack (Japan)”.

ROM: each of the files contained in a *set*.

Example: In the game “1941 - Counter Attack”, the *main set*, “1941 - Counter Attack (World)”, contains the following *ROMs*:

41e_30.rom
41e_31.rom
41e_35.rom
41e_36.rom
41_09.rom
41_18.rom
41_19.rom
41_32.rom
41_gfx1.rom
41_gfx3.rom
41_gfx5.rom
41_gfx7.rom

ROM set: self-explanatory; groups of *ROMs* that make a particular set.

Example: The previous table is the *ROM set* for the *set* “1941 – Counter Attack (World)”.

Usually, *ROM sets* are contained within *ZIP* files. Each *set* must be in an independent file (we will see later that this is not strictly necessary) with a particular name. Continuing with our example, we can say that the two 1941 *sets* must be named 1941.zip (World) and 1941j.zip (Japan).

ROM sets can also be stored in directories. In this case, each *set* must be placed in an individual directory, where the name of the directory must be the same as the *ZIP* file.

We are now going to see in more detail the different possibilities we have to store *ROM sets*.

2. SPLIT / MERGE. What does this mean?

MAME can store *ROM sets* in three different ways:

- *Split / Not Merged*
- *Split / Merged*
- *Not Split*

In order to explain the differences, we will use the previous example: “1941 – Counter Attack”. Keep in

mind the following table which shows the *ROMs* for each of the two *sets*:

1941 - World (main set)	1941 - Japan (clone set)
41e_30.rom	4136.bin
41e_31.rom	4137.bin
41e_35.rom	4142.bin
41e_36.rom	4143.bin
41_09.rom	41_09.rom
41_18.rom	41_18.rom
41_19.rom	41_19.rom
41_32.rom	41_32.rom
41_gfx1.rom	41_gfx1.rom
41_gfx3.rom	41_gfx3.rom
41_gfx5.rom	41_gfx5.rom
41_gfx7.rom	41_gfx7.rom

2.1. Method 1: Split / Not Merged

In the *Split / Not Merged* mode, there is an independent file for each *ROM set*, each containing all the required *ROMs* for the *set*. For our example, our two files would be: 1941.zip (containing de 12 *ROMs* in the left table column) and 1941j.zip (containing the 12 *ROMs* from the right column).

2.2. Method 2: Split / Merged

By examining the table it can be seen that the last 8 *ROMs* are identical for both sets. Storing the *ROMs* in *Not Merged* means that there will be unnecessary duplication of information, which just wastes storage space. MAME can solve this problem by erasing the duplicated *ROMs* from each of the *clone sets* and storing them only in the *main set*. This way the *ROM sets* in our example would look like:

1941 - World (main set)	1941 - Japan (clone set)
41e_30.rom	4136.bin
41e_31.rom	4137.bin
41e_35.rom	4142.bin
41e_36.rom	4143.bin
41_09.rom	
41_18.rom	
41_19.rom	
41_32.rom	
41_gfx1.rom	
41_gfx3.rom	
41_gfx5.rom	
41_gfx7.rom	

This method reduces the size of the file 1941j.zip without any loss of information, because the *ROMs* we have erased are kept in the *main set*. This method is known as *Split / Merged*.

2.3. Method 3: Not Split

There is one final method: *Not Split*. If there was an independent file for each set, for MAME version

0.59 there would be a total of 3290 files (the number of *sets* for v0.59). To reduce that number, *ROMs* of different *sets* of a game can be stored in a single file (the *main set*). In our example, the file would contain:

1941 - World (<i>main set</i>)	1941 - Japan (<i>clone set</i>)
41e_30.rom	
41e_31.rom	
41e_35.rom	
41e_36.rom	
41_09.rom	
41_18.rom	
41_19.rom	
41_32.rom	
41_gfx1.rom	
41_gfx3.rom	
41_gfx5.rom	
41_gfx7.rom	
4136.bin	
4137.bin	
4142.bin	
4143.bin	

The *main set* has not only its *ROMs* but also includes the unique *ROMs* of the *clone set*. This way, there is no need to keep the file 1941j.zip as it is empty.

In this example only one single file is removed. When you take into consideration that some games have lots of different versions (for example, Pac-Man, Galaxian or Asteroids), we have reduced the number of files from 3290 down to around 1870 – the number of games MAME supports (for version 0.59).

2.4. Summary

In summary, the three different storing modes supported by MAME are:

- *Not Split*: One single file for each game with all the *ROMs* of the *main set* and the *clone set*.
- *Split*: One file for each set.
 - *Not Merged*: All the *ROMs* in all the *sets*.
 - *Merged*: Only the unique *ROMs* in the *clone sets*.

2.5. Final Considerations

Is merging really useful to save space, or are we making it more complicated just for the sake of saving some disk space?

A good example of the huge space saving that merging can bring is the game “Dungeons & Dragons: Shadow Over Mystara”. In MAME v0.59, this game has 6 *sets* (this number can vary with future versions). In *Split / Not Merged* mode, each *set* sizes 33.3 MB but in *Merged* mode, 28 MB of redundant *ROMs* is deleted for each of the *clone sets*. This represents a total saving of approximately 140 MB!

Considering that the complete *ROM set* in *Split / Merged* mode for MAME 0.59 is stored in 4 GB, and taking into account this example, we can say that we are saving a couple of GB of space by not using the *Split / Not Merged* mode.

The only real drawback of merging is that the missing redundant *ROMs* in *clone sets* means loss of compatibility with other emulators. If other emulators are to be used, we should keep our *ROM sets* in *Split / Not Merged* mode.

3. UPDATING ROMS

3.1. Why?

If you've got a full set of *ROMs* for a specific version of MAME, then you'll only need to obtain *ROMs* for the newer games when a newer version of MAME is released, right? Wrong. Unfortunately, it's not that easy. For different reasons, *ROM sets* are constantly being changed - ie, the contents of the ZIP file may differ for each release.

One of the major modifications has been the renaming for all the NEO-GEO *ROMs*. If we had to make these changes manually, we'd have to rename *hundreds* of *ROMs* contained in ZIP files. Not difficult, but pretty time-consuming.

But things aren't as bad as all that. Fortunately, there are some programs that make this changes almost automatically, called ROM Managers. The following tutorial illustrates how to update MAME *ROM sets* using one of the best-known ROM managers: ClrMame Pro.

3.2. What do I need?

To update our MAME *ROM set* we need the following:

- Complete *ROM set* of an old version
- *Fixfiles*
- *Dat file* of the current MAME version.
- *ROM Manager*

3.3. What is a *fixfile*?

Fixfiles are files that contain *ROMs* needed to update from one version to the next. This includes two types of *ROMs*: *ROMs* for the new games supported, and *ROMs* added to previously supported games.

Why is it necessary to add *ROMs* to previously emulated games?

The MAME Team changes the source code from version to version. This means that some times we have to add or remove *ROMs* from a game due to some functionality issues.

3.4. ¿Qué es un *dat file*?

Dat files are text files with information about emulated games for a particular MAME version. In these files is information about *ROMs* contained in particular *sets*: names, merging information, which is the *main set* and which are *clones*, CRC for each *ROM*, etc. CRC is an algorithm used to check the integrity of a file, and is the system used by ROM Managers to identify the *ROMs* in order to verify their name, size and integrity.

3.5. Where can I get *dat files*?

Links to *dat files* can be found in almost every web page related to emulation. The most important ones (in my opinion) are the following:

<http://www.logiqx.com/> - Information, utilities and a big amount of *dat files*.

<http://opothspants.free.fr/rmd/> - If you don't find the *dat file* you are looking for here (or in one of the links), then it doesn't exist.

<http://www.clrmame.com/> - *dat files* to use with ClrMame Pro

We will see later that we can obtain MAME *dat files* from the executable file of the emulator itself.

3.6. What is a Rom Manager?

As we have mentioned previously, ROM Managers are programs that make our life easier when we have to deal with MAME *ROM* updates. But they are not just limited to MAME - they can be used with any other emulation system, as long as a *dat file* exists (and there are lots of them). It can even handle MAME32 add-ons (icons, cabinets, flyers, marquees and the like).

3.7. Where can I get Rom Managers?

In this tutorial we are going to explain the use of ClrMame Pro. You can find the latest version in the official site:

<http://www.clrmame.com/>

4. ROM UPDATE USING CLRMAME PRO

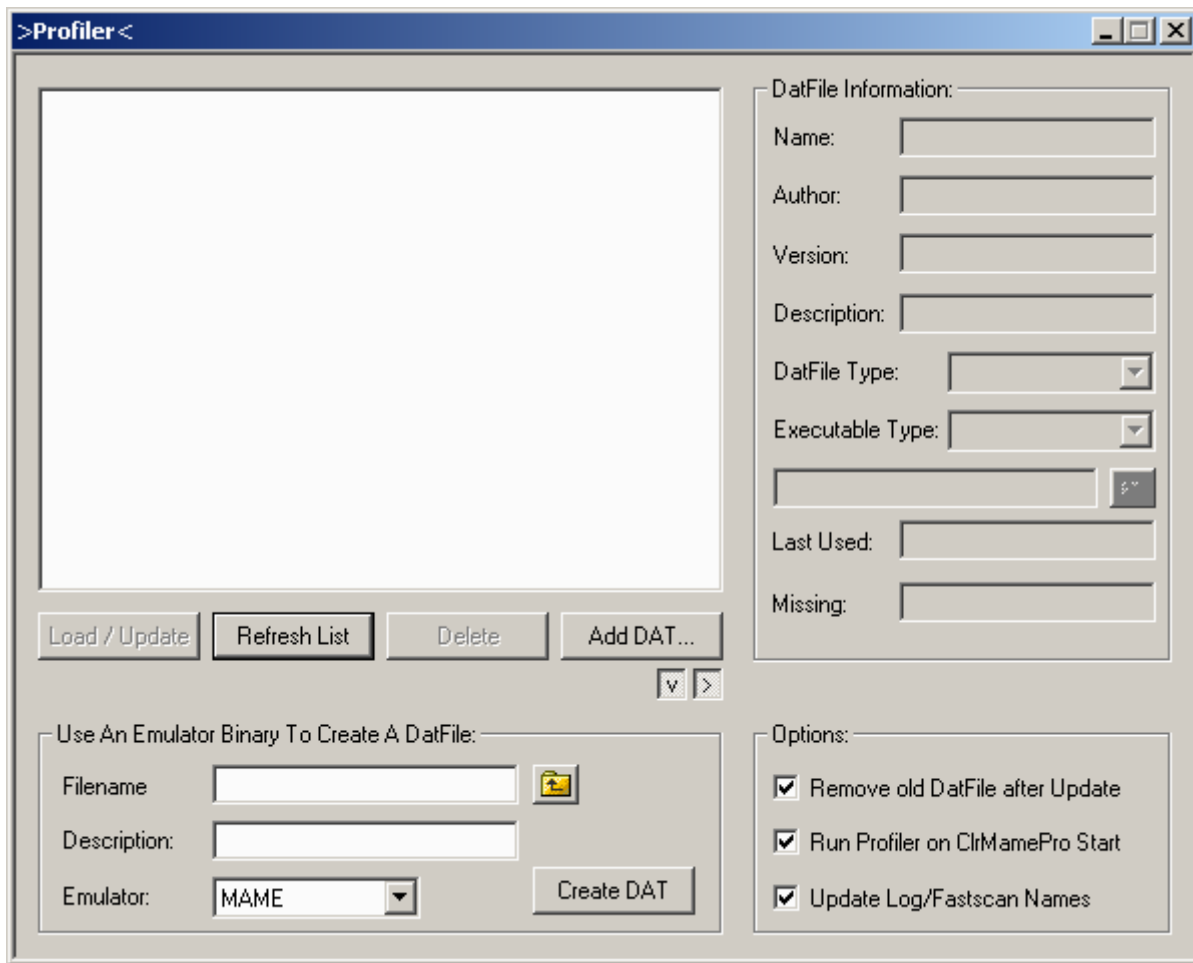
Note that this is a tutorial about a very specific use of the program. It is recommended to fully read the documentation in order to learn more about the full functionality of the program.

To update our *ROM set* using ClrMame Pro we must follow these steps:

1. Selecting the *dat file*
2. Loading the database
3. Configuring the *ROM* directories
4. Configuring the scanning options
5. Updating

4.1. Selecting the *dat file*

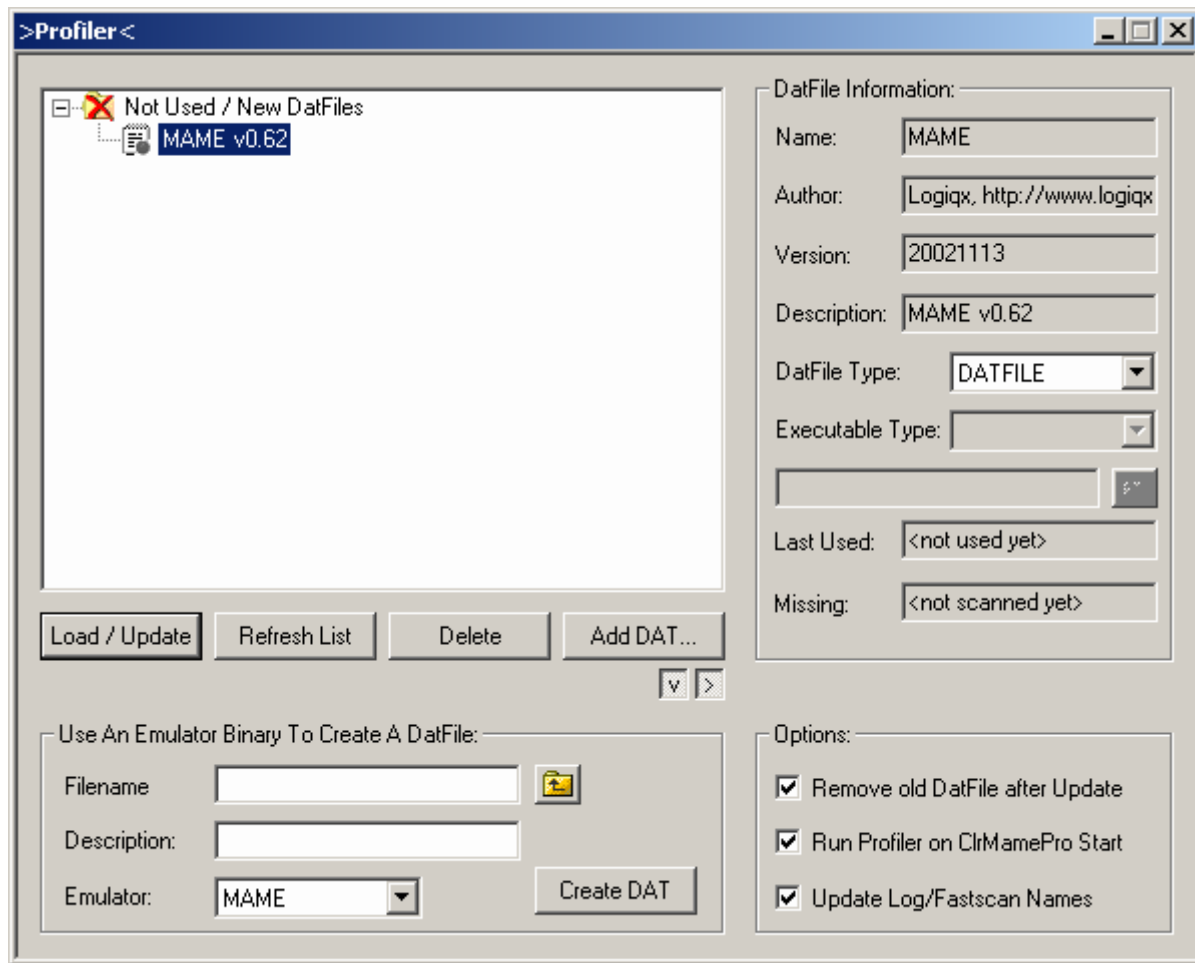
First, run the program. The "Profiler" window pops-up.



We have two *dat file* types to select: a previously downloaded *dat file* or a self-created *dat file* using the MAME executable file.

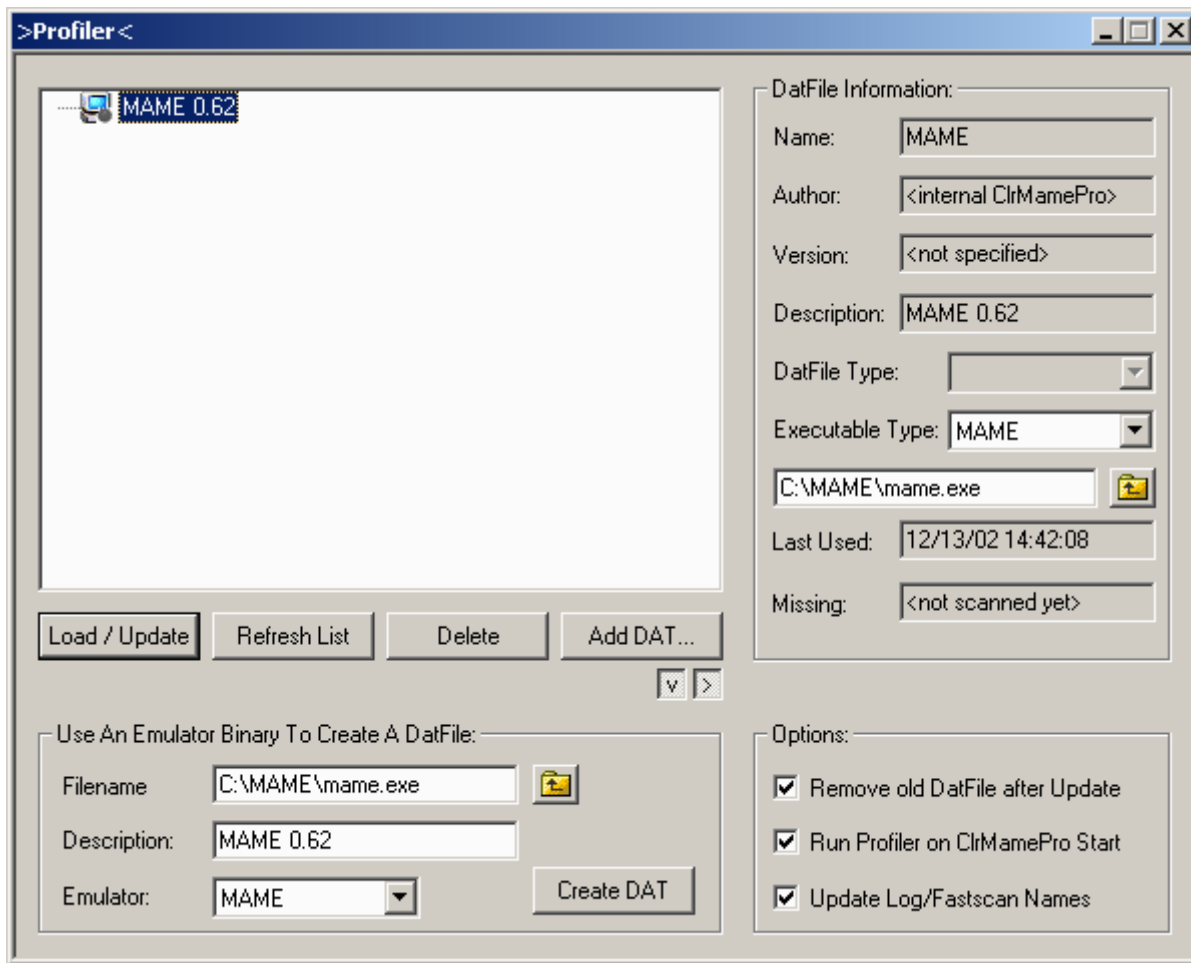
4.1.a. Previously downloaded dat file

In the “Profiler” window press Add DAT... and select the file containing the *dat file* (no need to unzip it). Once it is selected, the new *dat file* will appear in the upper left box in the “Profiler”.



4.1.b. Exe-Based *dat* file

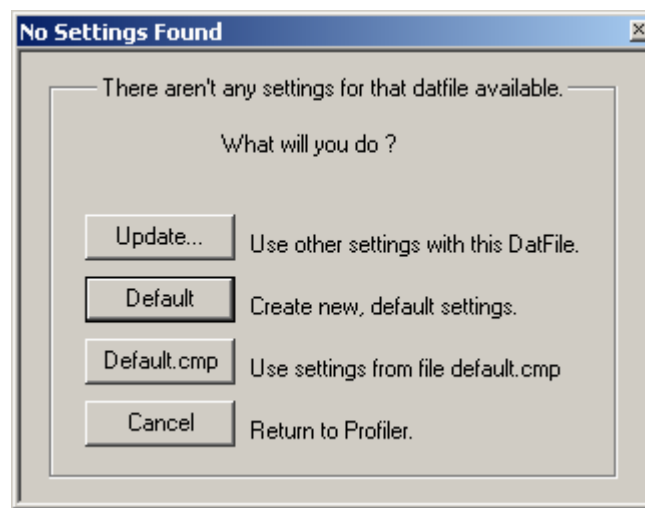
In the box Use An Emulator Binary To Create A DatFile, in Destination select the MAME executable file (in this case, it must be unzipped). In Description enter, for example, “mame 0.62”, we make sure that in Emulator we have MAME selected and press Create DAT. The new dat file will appear in the upper left box in the “Profiler”.



4.2. Loading the dat file

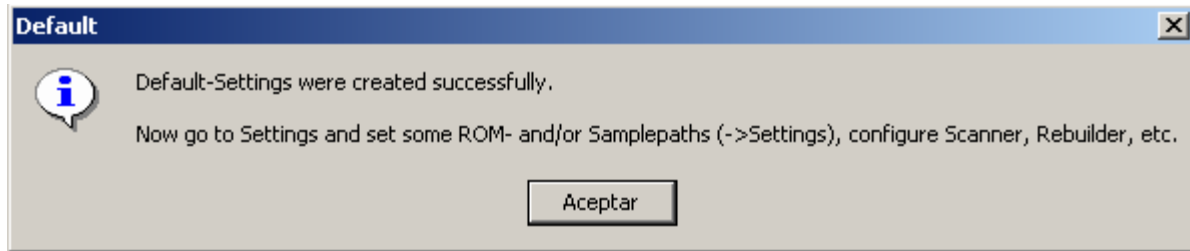
To load the *dat file* added in the “Profiler”, double-click on it.

If we have added a previously downloaded *dat file*, the following window will pop-up:



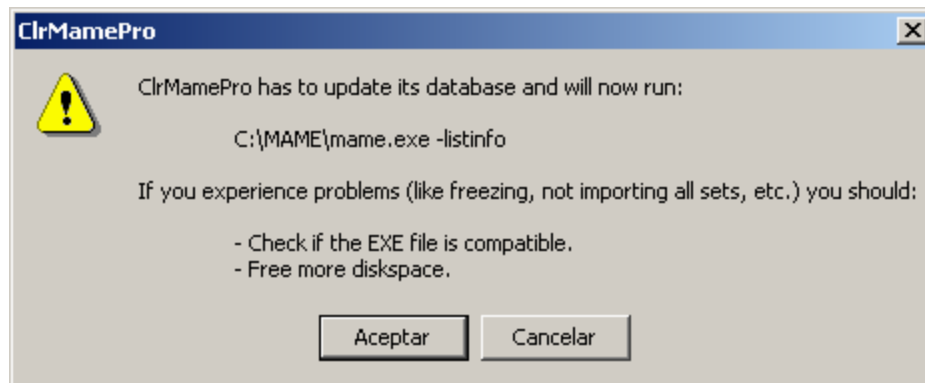
Press Default.

In the following window:



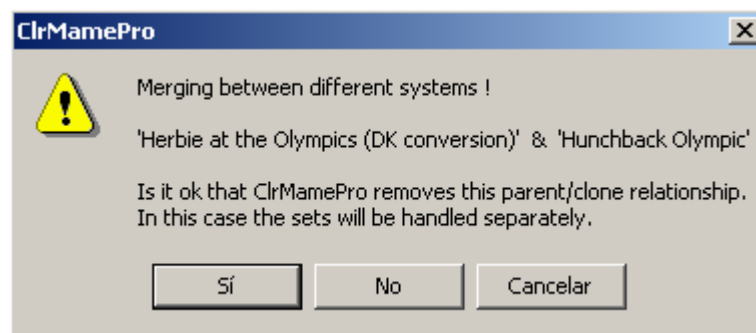
press Accept.

If we have created an exe-based *dat file*, the window that pops-up will be:



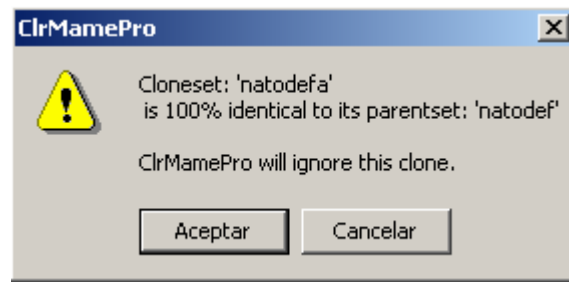
Press Accept.

Then, the load process begins, during which a series of warnings will be presented:



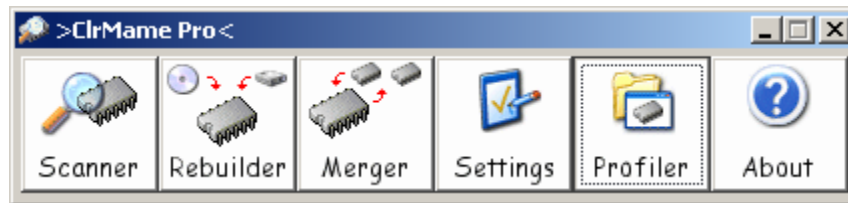
Press Yes.

In the final step of the process, you will see the following message:



Press Accept.

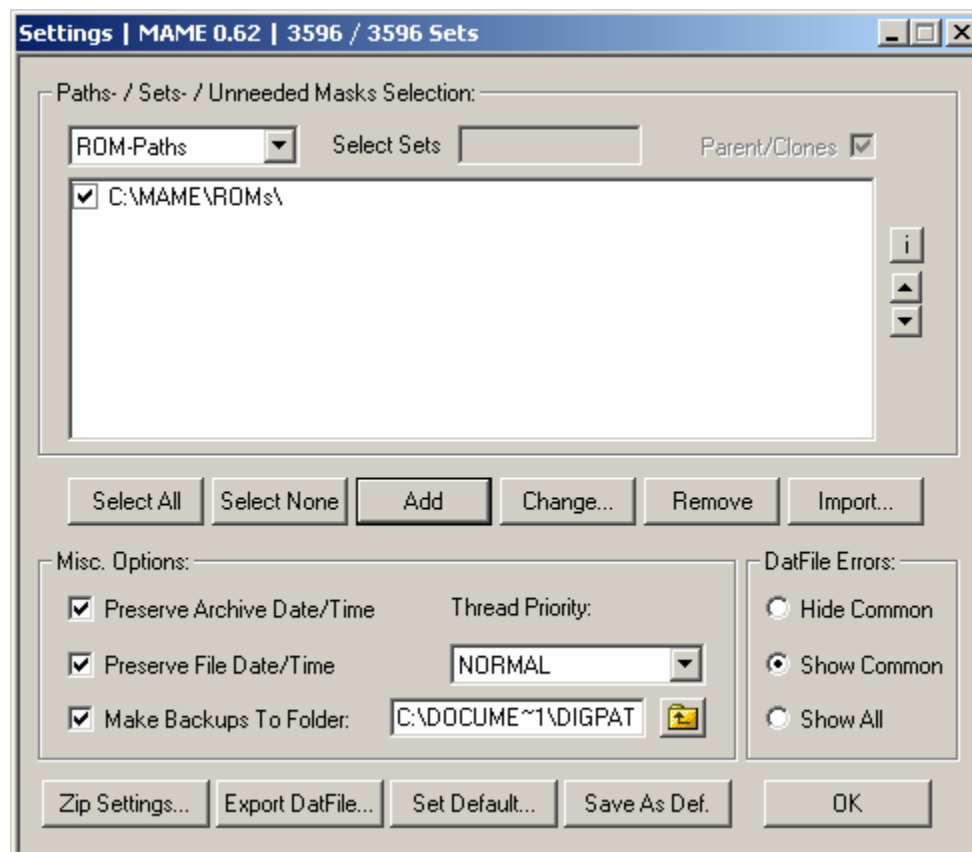
When the *dat* file is loaded, the main window pops-up.



4.3. Configuring the *ROM* directories

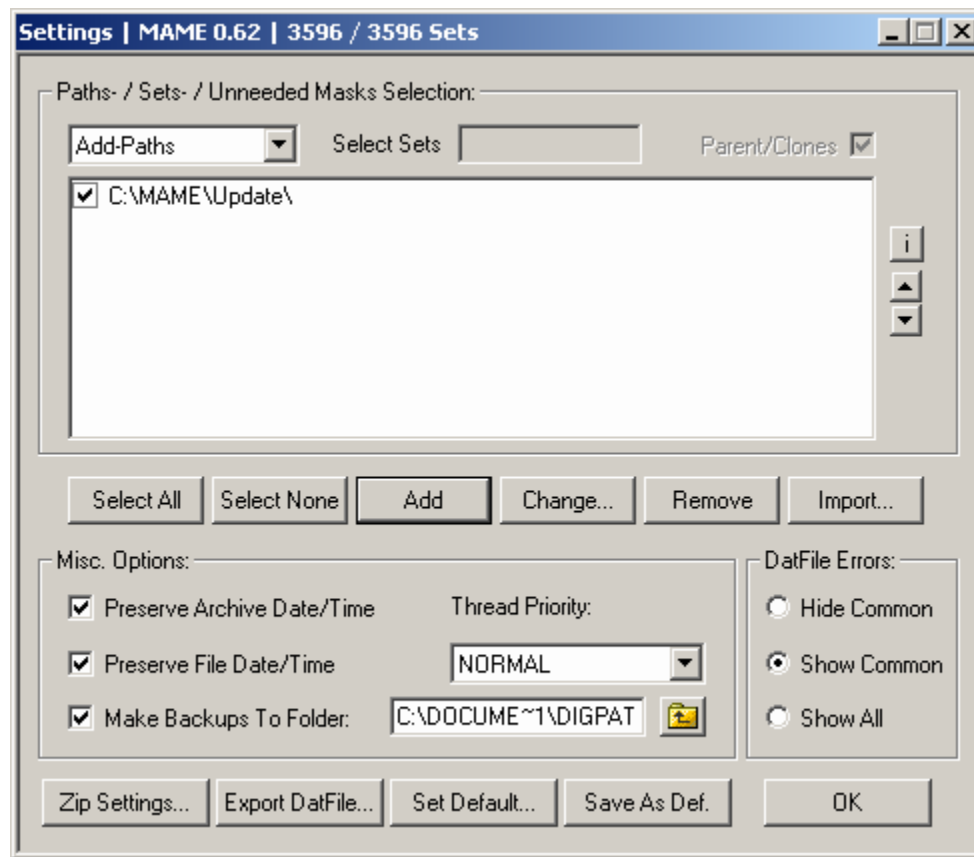
We must now tell the program where are we storing the *ROMs* and the *fixfiles*.

To add the *ROM* directories, press Settings.



Select ROM-Paths in the upper-left corner list. Press Add and select the directories where you are storing your MAME ROM set.

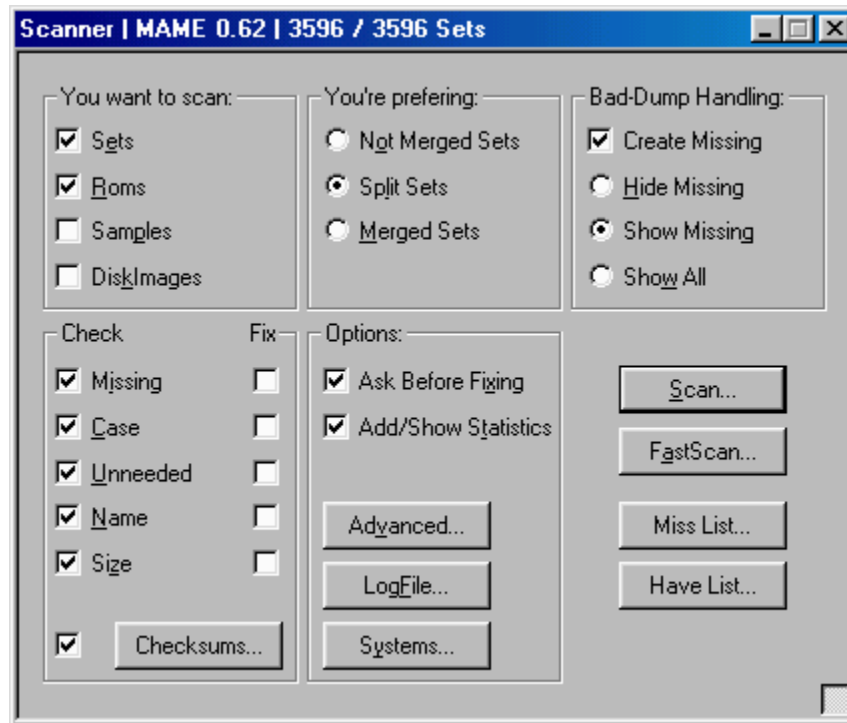
Select then Add-Paths in the list and add every *fixfiles* directory.



Once you have added every needed directory, press OK.

4.4. Configuring the scanning options

In the main windows press Scanner:



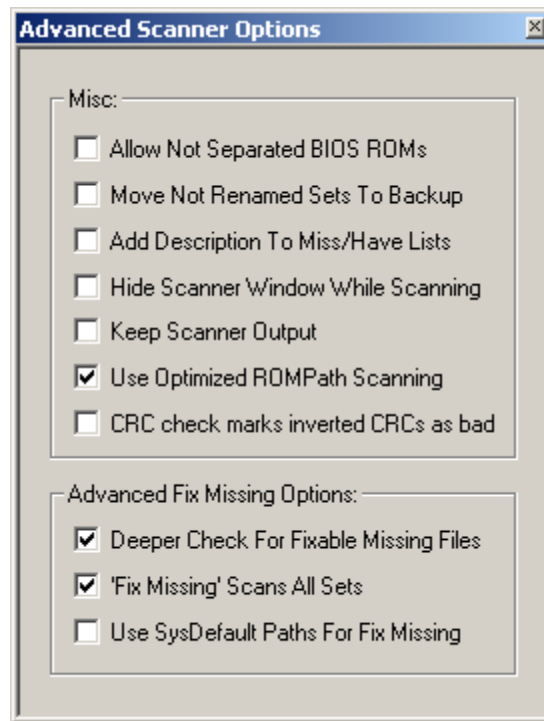
In the box You want to scan: check Sets and Roms and uncheck the rest.

In the box You're Using: select the *ROM set* type using the following equivalences:

You're Using	ROM set
Not Merged Sets	Split / Not Merged
Split Sets	Split / Merged
Merged Sets	Not Split

In the box Bad-Dump Handling check Create Missing and select Show Missing.

In the box Options press Advanced...:



In the box in the bottom, check Deeper Check For Fixable Missing Files and “Fix Missing” Scans All Sets.

Close this windows.

4.5. Updating

With the selected options, if we press Scan..., the program will only look for the required files in the ROMs directories we have selected in the Settings window, without performing any update operation. The results of the scanning appear in the white window next to the “Scanner” window.

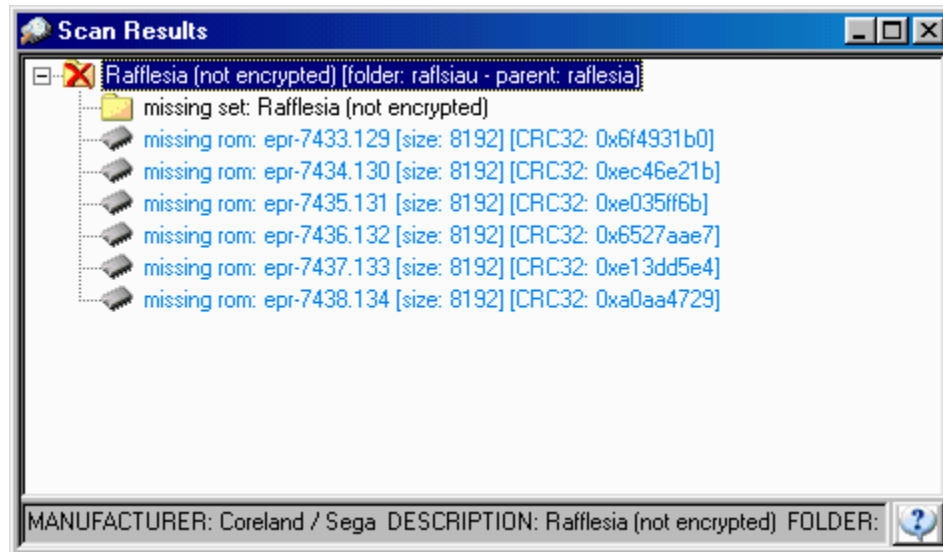
To update the files, check the boxes in the Fix column and press Scan.



Depending on the number of operations required and the equipment used, the updating process may take from a few seconds to several hours.

When the process is finished, uncheck the boxes in the Fix column and press again Scan. If the update has been performed correctly and we had all the required files, once the scan is finished there won't be

any message in the results window, or maybe a report about a missing game, “Rafflesia (not encrypted)”. There are no valid ROMs for this game. If there is no “Missing” report, we must know that the ROMs we have for this game have been artificially created to match the size and CRC requirements, but are not the real *ROMs* required to play the game.



5. ADDITIONAL INFORMATION

5.1. BIOS files

The best way to understand what a *BIOS* file is, is through an example. Let’s take a look at the NEO-GEO games. All of them require the following 4 *ROMs*:

neo-geo.rom
ng-sm1.rom
ng-lo.rom
ng-sfix.rom

Having these 4 *ROMs* in every single NEO-GEO *set* means having a lot of redundant files, so it was decided to store them in a separated file, what would be called a *BIOS* file, and when running one of the NEO-GEO games, MAME will look for those *ROMs* in the *BIOS* file. This file is neogeo.zip

In MAME 0.62, there are 6 *BIOS* files: cvs.zip, decocass.zip, neogeo.zip, pgm.zip, playch10.zip and skns.zip

The problem with these type of files is that ClrMame Pro doesn’t update them so when after a scanning process we have some *BIOS* errors, we must correct them manually.

The most common error arises when in a new MAME release is included a new *BIOS* file. When we update our *ROM set*, this new file won’t be automatically copied to the *ROMs* folder so if after the update we scan our *ROM set*, the Scanner would report that *BIOS* file as missing. To solve this, we must copy the file in the *ROMs* folder (the most recent example of this problem is the update from MAME v0.61 to v0.62, where the new skns.zip *BIOS* file has been included).

Another possible error is the renaming of some of the *ROMs* inside a *BIOS* file. We must again

manually perform the update by changing the name to the correct one (this has happened with some of the *ROMs* in the negeo.zip *BIOS* file).

5.2. *Fixfile* creation: mamediff and the Rebuilder

Imagine you have a complete MAME *ROM set* (for example v0.62) and you want to send to a friend the update CDs required to update his v0.60 *ROM set*. One possible way to do this is sending him the full *set*, but there is another way in which you would only be sending him the files required to perform the update (in his v0.60 *ROM set* he has almost everything required for v0.62, so no need to send him the full *set*).

To do this, we are going to produce a *dat file* for the required files, and then we will create the files using our complete *ROM set*.

5.2.a. Dat file creation

We will need a small utility called mamediff, whose author, Logiqx, has available in his web site:

<http://www.logiqx.com/>

First, we must get the exe files for the MAME versions we are going to use (in our example, v0.60 and v0.62). These are available in the MAME official site:

<http://www.mame.net/>

Once we have obtained these files, mame060.exe and mame062.exe (the exes must be renamed to avoid overwriting one with the other when unzipping them) we open a DOS box and type the following commands:

```
mame060 -listinfo > mame060.txt  
mame062 -listinfo > mame062.txt
```

This will produce two text files (quite big) with the information of the full *ROM set* of each MAME version, that mamediff will later use to create the *dat file*. We must copy these two text files to the mamediff folder.

Now you have to type one of the following commands (be careful with the capitals, as the program is case sensitive):

```
mamediff -N -T mame060.txt mame062.txt  
mamediff -M -T mame060.txt mame062.txt  
mamediff -S -T mame060.txt mame062.txt
```

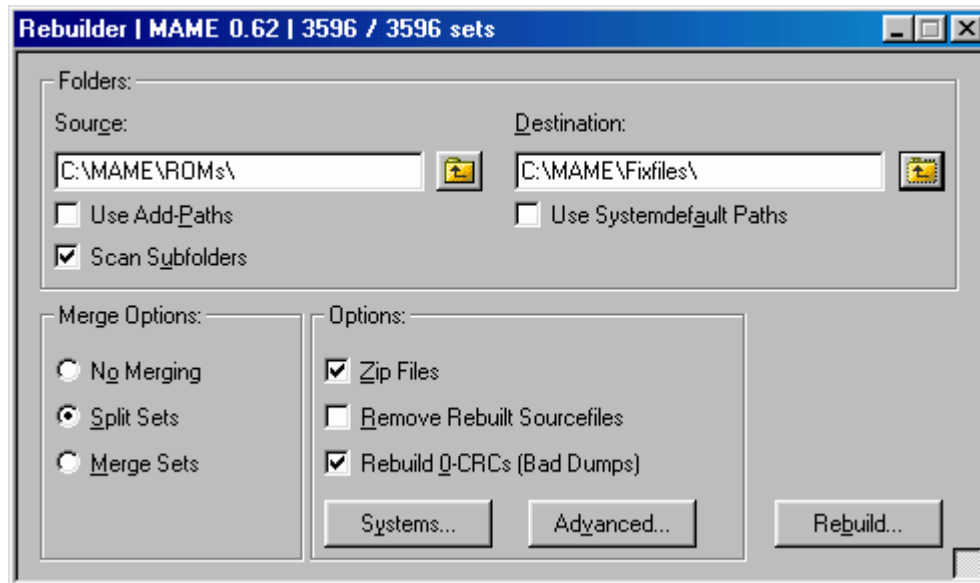
Option -N is for Split / Not Merged *ROM sets*, option -M for Not Split and option -S for Split / Merged.

This will generate the file mamediff.dat, which is the *dat file* we will now use with ClrMame Pro to create the required files to perform the update.

5.2.b. File creation

Open ClrMame Pro and load the *dat* file as explained in sections 4.1 and 4.2 of this guide.

In the Main Windows, press “Rebuilder”.



In Source, select the folder where you store your complete MAME v0.62 *ROM set*.

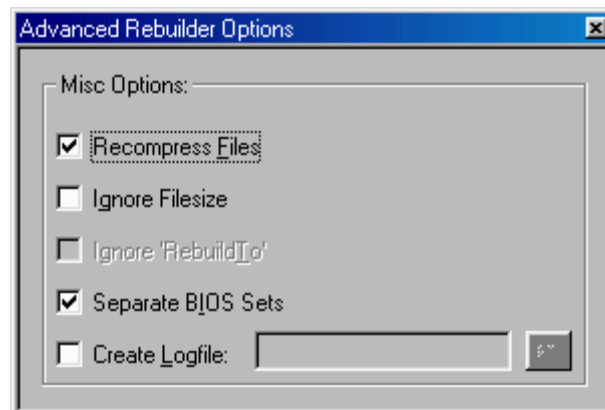
In Destination, the folder where you want to create the update files.

In Merge Options select the *ROM set* type using the following table:

Merge Options	Tipo de ROM set
No Merging	Split / Not Merged
Split Sets	Split / Merged
Merge Sets	Not Split

In Options, check Zip Files and Rebuild 0-CRCs (Bad Dumps).

Press Advanced... The following window pops up:



Check Recompress Files and Separate BIOS Sets and close the window.

Press Rebuild...

Once the process is done, you will have in the destination folder the files required to update from v0.60 to v0.62 (what we called *fixfiles* in sections 3 and 4). To perform the update itself, just go back to section 4 in this guide.

6. ACKNOWLEDGEMENT

First, I would like to thank Nicola Salmoria and the MAME team. Without them, this guide wouldn't have a reason to exist. Thanks for all the time and effort you spend in this great project we know as MAME.

Thanks to Roman Scherzer, ClrMame Pro author, for working in such a great utility and for giving his permission to let me include all the material related to his program in this guide.

For the same reason, thanks to Loqix, mamediff author, for giving his permission to let me use it in section 5 in this guide.

Also thanks to 'Dungeon' Dave for his help in translating most of this guide from Spanish to English.

Finally, my gratitude to Tom B, for some other spelling corrections in this English version, an especially for generating the PDF versions of this guide.

Many thanks to all of them.