


 Search

MESS User's Manual

Disclaimer: This manual is not complete yet. If you find any outdated or incorrect info, or if you think there are important aspects of the MESS Usage which are not covered by this guide, please post at the [MESS Message Board](http://forums.bannister.org/ubbthreads.php?ubb=postlist&Board=1) [<http://forums.bannister.org/ubbthreads.php?ubb=postlist&Board=1>].

Shortcuts

Getting Started	Console Emulation	Internal UI Options	Controller Options	System Specific Options
Common Issues	Computer Emulation		Cheat Options	MESS Tools
Folder & Files	Chess Computer Emulation		Save Options	MESS Error Messages
Command Line Options	Calculator Emulation	Video Options	Snapshot Options	
Software Lists	Media Emulation	Sound Options	Recording Options	

Getting Started

Shortcuts

MESS - An Introduction	Supported OS	Using MESS	Pause MESS	Quit MESS
Purpose of MESS	Installing MESS	Compiling MESS	Access MESS Internal UI	Frontends
Systems Emulated by MESS	Requirements	Start MESS		

MESS - An Introduction

MESS is an acronym which stands for Multi Emulator Super System. MESS documents and reproduces through emulation the inner components of computers, consoles, chess computers and calculators. As a nice side-effect, MESS allows to use on a modern PC those programs and games which were originally developed for the emulated machines.

MESS is mostly programmed in C with some core components in C++ ([here](#) you can see an overview of the MESS source structure), and shares its core and CPU emulation with MAME. This allows MESS to offer the same powerful interface and flexible options which MAME offers for arcade emulation.

MESS can currently emulate over 550 individual systems from the last 5 decades.

Purpose of MESS

The primary purpose of MESS is to preserve decades of computer and console history. As technology continues to rush forward, MESS prevents these important “vintage” systems from being lost and forgotten. MESS is based on MAME and shares many of its components. Additional information about MESS and MAME can be found at the [MESS wiki](http://mess.redump.net) [<http://mess.redump.net>] or at [MAMEDev](http://mamedev.org) [<http://mamedev.org>]. You could also find the [MAME FAQ](http://mamedev.org/devwiki/index.php?title=Frequently_Asked_Questions) [http://mamedev.org/devwiki/index.php?title=Frequently_Asked_Questions] and [MESS FAQ](#) a very interesting read.

Systems Emulated by MESS

[ProjectMESS](#) contains a complete list of the systems currently emulated. As you will notice, being supported does not always mean that the status of the emulation is perfect. You may want

1. to check the status of the emulation in the wiki pages of each system, accessible from the [drivers page](#) (e.g. for Apple Macintosh, from the page for the [mac.c driver](#) you can reach the pages for both [macplus](#) and [macse](#)),
2. to read the corresponding [sysinfo.dat](#) entry in order to better understand which issues you may encounter while running a system in MESS (again, for Apple Macintosh Plus you have to check [this entry](#)).

Alternatively, you can simply see the status by yourself, launching the system emulation and taking a look to the red or yellow warning screen which appears before the emulation starts, if any.

Notice that if you have information which can help to improve the emulation of a supported system, or if you can directly contribute fixes and/or addition to the current source, you can follow the instructions at the [contact page](http://mess.redump.net/contacts.php) [<http://mess.redump.net/contacts.php>] or post to the [MESS Message Board](#) [<http://forums.bannister.org/ubbthreads.php?ubb=postlist&Board=1>].

Supported OS

As with MAME, the current source code can be directly compiled under all the main OSes: Windows (both with DirectX native support and with SDL support), Linux, FreeBSD, Mac OSX and OS/2 (these latter Operative Systems only requiring SDL 1.2 to work). Also, both 32-bit and 64-bit executable can be built with no difficulties.

Installing MESS

Windows

You simply have to download the latest binary archive available at [this page](http://mamedev.org/release.html) [<http://mamedev.org/release.html>] and to extract its content to a folder. You will end up with many files (below you find explanations about some of these), and in particular with mess.exe. This is a command line program. The installation procedure ends here. Easy, isn't it?

Other Operating Systems

In this case, you can either look for pre-compiled (SDL)MESS binaries (e.g. in the repositories of your favorite Linux distro) which should simply extract all the needed files in a folder you choose, or compile the source code by yourself. In the latter case, see the [entry on compiling](#).

Requirements

MESS is written in fairly generic C/C++, and has been ported to numerous platforms. Over time, as computer hardware has evolved, the MESS code has evolved as well to take advantage of the greater processing power and hardware capabilities offered.

The official MESS binaries are compiled and designed to run on a standard Windows-based system. The minimum requirements are:

- Any MMX-capable AMD or Intel processor (Pentium III or later recommended for current versions)
- Windows 98 or later (Windows 2000 or later preferred)
- DirectX 5.0 or later (included with all versions of Windows 98 or later)
- A DirectDraw or Direct3D capable graphics card
- Any DirectSound capable sound card

Of course, the minimum requirements are just that: minimal. You may not get optimal performance from such a system, but MESS should run. Modern versions of MESS require more power than older versions, so if you have a less-capable PC, you may find that using an older version of MESS may get you better performance, at the cost of lowered accuracy and fewer supported systems.

As of MESS 0.106 and later, MESS will take advantage of 3D hardware for compositing artwork and scaling the games to full screen. To make use of this, you should have a modern Direct3D 8-capable video card with at least 16MB of video RAM.

Around the same time, MESS added minimal multi-processor support, if you use the -mt flag. This means that some of the video processing can be done on a second CPU core if it is available. To take advantage of this, you should run MESS on a dual core (or greater) system.

Keep in mind that even on the fastest computers available, MESS is still incapable of playing some systems at full speed. The goal of the project isn't to make all system run speedy on your system; the goal is to document the hardware and reproduce the behavior of the hardware as faithfully as possible.

(from mamedev.org, adapted)

BIOS Dumps

Most of the systems emulated by MESS requires a dump of the internal chips of the original system. These can be obtained by extracting the data from an original unit, or finding them (at your own risk) in the WorldWideWeb. Being copyrighted material, MESS does not come with any of these.

Also, you may want to find some software to be run on the emulated machine. Again, [Google](http://www.google.com) [<http://www.google.com>] and other search engines are your best friends. MESS does not provide any software to be run on the emulated machines because it is very often (almost always, in the case of console software) copyrighted material.

Using MESS

If you are a new MESS user, you could find this emulator a bit complex at first. The basic usage, from command line, is

```
mess.exe <system> <media> <software> <options>
```

where

- <system> is the shortname of the system you want to emulate (e.g. nes, c64, etc.)
- <media> is the switch for the media you want to load (if it's a cartridge, try -cart or -cart1; if it's a floppy disk, try -flop or -flop1; if it's a CD-ROM, try -cdrom)
- <software> is the program / game you want to load (and it can be given either as the fullpath to the file to load, or as the shortname of the file in our [software lists](#))
- <options> is any additional [command line option](#) for controllers, video, sound, etc.

Remember that if you type a <system> name which does not correspond to any emulated system, MESS will suggest you some possible choices which are close to what you typed; and if you don't know which <media> switch are available, you can always launch

```
mess.exe <system> -listmedia
```

If you don't know what <options> are available, you have many possibilities. First of all, you can check the [command line options](#) section of this Manual. Moreover, since MESS shares most of its core components with MAME (and its Windows GUI with MAMEUI), any guide for MAME would fit perfectly for MESS as well, giving you directions about MESS options.

Alternatively, you should keep in mind the following command line options, which might be very useful in this context: `help`, `showusage` (or briefly `su`), `showconfig` (or briefly `sc`) and `createconfig` (or briefly `cc`).

If you launch at command line

```
mess.exe -help
```

you will be explained what MESS is the basic structure of MESS launching options, i.e. as explained above

```
mess.exe <system> <media> <software> <options>
```

If you launch at command line

```
mess.exe -showusage
```

you will be prompted with the (quite long) list of available command line options for MESS. The main options are described below, in the [Command Line Options](#) section of this manual. You can also read all the command line options at the [usage](#) page of this wiki.

If you launch at command line

```
mess.exe -showconfig
```

you will be prompted with the (quite long) list of available configuration options for MESS. These configuration can always be modified at command line, or by editing them in `mess.ini` which is the main configuration file for MESS. You can find a description of some configuration options in the [Command Line Options](#) section of this manual (usually, to each configuration option there corresponds a command line option to configure and modify it). You can also read all the configuration options at the [mess.ini](#) page of this wiki.

Finally, if you launch at command line

```
mess.exe -createconfig
```

you will create a brand new [mess.ini](#) file, with default configuration settings. Notice that `mess.ini` is basically a plain text file, hence you can open it with any text editor (e.g. Notepad, Emacs or TextEdit) and configure every option you need. However, no particular tweaks are needed to start, so you can basically leave most of the options unaltered the first times you use MESS.

[maybe the -soft opengl option for (SDL)MESS should be mentioned here...]

Once you are more confident with MESS options, you may want to configure a bit more your setup. In this case, keep in mind the order in which options are read:

1. MESS starts with all the default settings
2. MESS overwrites the modified options found in .ini files, if any; in particular MESS reads in order
 - `mess.ini` found in the same directory of the emulator, if any
 - `mess.ini` again (so that, if you specified an alternative path for .ini files in the main `mess.ini`, this new location will be checked)
 - `driver.ini` (so that, if you are interested e.g. in msx clones, you can modify all their options at once by using a single `msx.ini` configuration file)
 - `parent.ini` (so that...)
 - `system.ini` (so that..)
3. MESS overwrites the modified options passed to command line

Therefore, options set at command line have precedence over anything you have set in .ini files, while unspecified options fall back to their default values.

TO DO: check if the ini part is 100% correct and add explanations about parent and system .inis

Compiling MESS

MESS / MESSUI (Windows)

Compiling MESS for Windows requires the MinGW toolkit, a variant of GCC. MinGW comes with a variant of GDB for debugging. Downloads of those tools can be found on the [Build Tools page](#). Notice also that the same tools are used to compile MAME. Hence, you can download them from the correspondent [MAME Tools for building MAME on Windows page](#) [<http://mamedev.org/tools/>] as well.

Then, you need the MESS source, of course. You can either take the whole tree from the svn repository (see [this page](#)), or download the latest stable source package (see [this page](#) instead). The SVN is, however, the recommended choice because of the frequent updates.

Once you have the source, to compile the command line version of MESS is enough to open a command prompt and type (assuming MinGW has been correctly installed as explained at [MAME Tools for building MAME on Windows page](#) [<http://mamedev.org/tools/>])

```
make TARGET=mess
```

Please notice the capital letters for TARGET: if you use target=mess it won't work.

Other possible compile options are the following

```
DEBUG=1
SYMBOLS=1
```

The former enables stricter validation inside the emulator and some additional logging (but notice that the basic debugger is compiled even without this option, and it is hence available in **all** builds). The latter enables symbols in the compile, to make easier the debugging of crashes through gdb.

Alternatively, you can use Visual C to compile MESS. Indeed, if you add MSVC_BUILD=1 at the compile command, the build will use Visual C instead of MinGW. Note that MinGW is still required, as GNU make is used rather than nmake. Using the Microsoft compilers allows to use the Visual C debugger to debug MESS, but one does not get the full benefits of the IDE.

If you want to compile MESSUI, then you need to type

```
make TARGET=mess OSD=winui
```

Finally, if you want to compile the Tools (e.g. chdman, imgtool and castool), then you need to type

```
make TARGET=mess tools
```

(if you forget the “TARGET=mess” part you will end up only compiling the MAME tools).

You can also read our [Compile MESS wiki page](#).

(SDL)MESS (Windows / *nix / Mac OSX / Other OS)

Starting from version 0.137, the SDL component of MAME/MESS are included in the official source.

Therefore, to compile MESS you only need to download the source from [this page](#), or to checkout the SVN repository (see [this page](#) for instructions about SVN).

Required packages and instructions about compiling (SDL)MESS are the same used to compile (SDL)MAME and can be found following the links at [this thread \[http://forums.bannister.org/ubbthreads.php?ubb=showflat&Number=35138#Post35138\]](http://forums.bannister.org/ubbthreads.php?ubb=showflat&Number=35138#Post35138). In general, you only need a compatible version of GCC (sometimes latest bleeding edge versions may cause compile issues), the SDL libraries and the MESS & MAME sources: the makefile has been written so that typing

```
make TARGET=mess
```

is enough to correctly compile the right version for your setup, by automatically choosing the appropriate OS and selecting between 32-bit and 64-bit architecture. The only case which require an additional option is if you want an SDL build in Windows. In this case, you have to type

```
make TARGET=mess OSD=sd1
```

Notice that, if you are using MacOSX 10.6, the default settings is to compile a 32-bit executable. At your own risk, you might try to compile MESS in 64-bit too by typing

```
make TARGET=mess PTR64=1
```

This is not officially endorsed, but it produces a slightly faster executable and there have been no issues reported so far.

Please notice the capital letters for TARGET: if you use target=mess it won't work.

Other possible compile options are the following

```
DEBUG=1
SYMBOLS=1
```

The former enables stricter validation inside the emulator and some additional logging (but notice that the basic debugger is compiled even without this option, and it is hence available in **all** builds). The latter enables symbols in the compile, to make easier the debugging of crashes through gdb.

Finally, if you want to compile the Tools (e.g. chdman, imgtool and castool), then you need to type

```
make TARGET=mess tools
```

(if you forget the “TARGET=mess” part you will end up only compiling the MAME tools).

You can also read our [Compile MESS wiki page](#).

Start MESS

To start your adventure in the MESS world, you simply have to double click on the executable. You will reach the Internal User Interface (see the [Internal UI Options](#) section below) and you will be asked to choose a system among the supported ones.

When launching MESS at the command line you can always use -help or -showusage options to obtain a list of available options. You can also read the [usage](#) page of this wiki, which lists the options as they are output when you use the -showusage option.

Pause MESS

To take a break from the emulation, simply press <key>P</key>. Notice that, if you are emulating a computer and you are in the fully emulated keyboard mode, <key>P</key> will have no Pause effect. In this case, press <key>ScrLock</key> (<key>Ins</key> for (SDL)MESS) to switch your keyboard in partial emulation mode and press again <key>P</key> to actually pause MESS. When you want to resume the emulation, simply press <key>P</key> once more.

You may want to advance frame by frame in the emulation, while in Pause. This can be accomplished by pressing <key>S-P</key> during Pause.

See [Key Functions](#) section for more details about controlling MESS.

Access MESS Internal UI

To enter the Internal User Interface during emulation, press <key>Tab</key>. Notice that, if you are emulating a computer and you are in the fully emulated keyboard mode, <key>Tab</key> won't have the expected effect. In this case, press <key>ScrLock</key> (<key>Ins</key> for (SDL)MESS) to switch your keyboard in partial emulation mode and press again <key>Tab</key> to actually enter the User Interface.

Tips - the internal UI is completely clickable with your mouse. You can select options and move through the available menu by a simple double click of your mouse!

See [Key Functions](#) section for more details about controlling MESS.

Quit MESS

To stop MESS emulation, press <key>Esc</key>. Notice that, if you are emulating a computer and you are in the fully emulated keyboard mode, <key>Esc</key> will have no Quit effect. In this case, press <key>ScrLock</key> (<key>Ins</key> for (SDL)MESS) to switch your keyboard in partial emulation mode and press again <key>Esc</key> to actually quit MESS. After pressing <key>Esc</key>, you will

- go back to internal UI, if you started mess.exe by double clicking the executable, or
- go back to Prompt / Bash / Terminal, if you started mess.exe from command line.

See [Key Functions](#) section for more details about controlling MESS.

Frontends

If you don't like using MESS from command line, you may want to use MESS in conjunction with a frontend. Various frontends currently support MESS. In alphabetical order:

- [EmuCon](http://emuloader.mameworld.info/emucon/) [http://emuloader.mameworld.info/emucon/] - Windows only
- [Gelide](http://gelide.sourceforge.net/) [http://gelide.sourceforge.net/] - Linux
- [GNOMESS](http://bobz38.free.fr/gnomess/index.php) [http://bobz38.free.fr/gnomess/index.php] - Linux
- [HyperSpin](http://www.hyperspin-fe.com/) [http://www.hyperspin-fe.com/] - Windows - Atari 5200 tutorial [http://hyperspin-fe.com/oldsite/index.php?option=com_content&view=article&id=104&Itemid=116]
- [kxname](http://sourceforge.net/projects/kxname/) [http://sourceforge.net/projects/kxname/] - Linux
- [MAME Plus! GUI](http://sourceforge.net/projects/mameplus/) [http://sourceforge.net/projects/mameplus/] - Windows, Linux and Mac OS X supported
- [MEAGRE](https://sites.google.com/site/meagrefrontend/home) [https://sites.google.com/site/meagrefrontend/home] - Windows only
- [MESS auto-launcher](http://eadmaster.altervista.org/pub/index.php?page=cliapps#mess) [http://eadmaster.altervista.org/pub/index.php?page=cliapps#mess] - Windows and *nix (command-line only)
- [MessMenu](http://macmess.org/) [http://macmess.org/] - Mac OS X only
- [MESSUI](http://messui.lemulation.com/) [http://messui.lemulation.com/] - Windows only (formerly part of the official project)
- [ML TNG](http://mameload.mameworld.info/) [http://mameload.mameworld.info/] - Windows only
- [My Arcade](http://robert.hurst-ri.us/myarcade/) [http://robert.hurst-ri.us/myarcade/] - Windows, Linux
- [QMC2](http://qmc2.batcom-it.net/) [http://qmc2.batcom-it.net/] - Windows, Linux and Mac OS X supported
- [Wah!Cade](http://www.anti-particle.com/wahcade.shtml) [http://www.anti-particle.com/wahcade.shtml] - Linux
- [xlbmessui](http://sourceforge.net/projects/xlbmessui/) [http://sourceforge.net/projects/xlbmessui/] - Linux
- [xMess GUI](http://code.google.com/p/xmessgui/) [http://code.google.com/p/xmessgui/] - cross-platform

Please refer to their home pages and support forums for more information about the available options of the frontends.

Common Issues

Shortcuts

BIOS Handling	Run Systems with Software	I Found an Issue!
Software Handling	Reset a System	Key Functions
Start a System	Change System	Tips

BIOS Handling

As mentioned in the [Requirements](#) section, to run most of the supported machines in MESS, you need BIOSes files, i.e. files which represent an exact copy of the content of chips present inside the real systems. Once you have these files, you can

- either zip them up in a file called <system>.zip and put them in the roms/ folder (where <system> is the machine you want to emulate, e.g. you need a msx.zip file in the roms/ directory for the ASCII MSX to work)
- or create a system/ subfolder in the roms/ folder and put the uncompressed files into this subfolder (again *system* is the machine you want to emulate, e.g. you need a roms/msx/ subfolder to store your uncompressed MSX BIOS)

Notice that you can modify the location of the BIOSes by setting the -rompath option (see [Command Line Options](#)) to something different than roms/ . You can also setting it to roms/ **and** a second folder of your choice.

Tips - MESS can load zipped BIOSes based on their CRC checksum. Hence, if you have a correct BIOS zipped in the roms/ folder, the names of the single files are not really important. However, if you keep your BIOS unzipped, MESS cannot load the files by CRC and you need the BIOS files to be named in a very precise way. If you keep receiving a missing files error message and you think you have already the required files unzipped in a subfolder of the roms/ folder, check if their names are correct.

Software Handling

There exists no hardcoded software locations in MESS and you can configure MESS to launch software from any directory you want.

If you are running MESS from command line, you simply have to specify the full path to the software. E.g. to launch a Sharp X1 floppy disk that you have in "E:\sharp x1\disk.d88", you just have to type

```
mess.exe x1 -flop1 "E:\sharp x1\disk.d88"
```

Notice the quote around the path, to properly handle the space in the folder name.

While running MESS, you can also change software by entering the Internal UI (press <key>Tab</key>), choosing File Manager, selecting the device you want to use (Cartslots, Cassettes, Floppy Disks, Cdroms, etc.) and browsing your folders until you find the desired software image.

Tips - In the File Manager, the navigation through folders filled of hundreds of files can be made a lot faster, if you know a couple of features of the Internal UI! First of all, <key>PgUp</key>, <key>PgDn</key>, <key>Home</key> and <key>End</key> are fully supported and allow for faster scroll of the list of filenames. Moreover, as of MESS 0.142, you can type the beginning of a filename, and the File Manager will scroll until the file with the closest name. Adding the fact that you can click with the mouse of the file to start it, finding the file you want to launch has never been so easy!

Start a System

If you are running MESS from command line, you have to type

```
mess.exe <system>
```

where `system` is the MESS system name (a string of characters identifying the system). If you type the wrong system name, you will be prompted by a list of approximate matches. While running MESS, you can also change emulated system by entering the Internal UI and choosing `Select New System`.

Notice that some systems require a software image to run (e.g. some consoles which cannot work without a cartridge image inserted). These systems will give an error when you try to start them from the Internal UI.

Run Systems with Software

In addition to the BIOS images, to fully enjoy emulation of a system you will need also some software. Again, this is copyrighted material and it is up to you to find it. However, once you have found it, you can load it in MESS by simply pointing the emulator to the right directory, e.g. to run Atari 2600 with an image of the cart `Frogger` in `.a26` format, you need to launch

```
mess.exe a2600 -cart "C:\path_to_image\frogger.a26"
```

Some systems (usually consoles) do not start in MESS without an image (which would correspond to a cartridge inserted in the cartslot of the console). Other systems (e.g. all the computers) would start also without any software mounted. As an example, running

```
mess.exe c64
```

would prompt you to the BASIC “ready” screen. Different systems supports in MESS different types of images and different media. You can check supported media devices and file extensions by using the `-listmedia` option (also described [here](#)). E.g.

```
mess.exe apple2gs -listmedia
```

for specific Apple IIgs emulation or

```
mess.exe -listmedia
```

for a list of all supported media and file extensions for each system. You can always change the software you have loaded in a device by entering the Internal UI (pressing `<key>Tab</key>`) and selecting “File Manager” (see [below](#)).

As of MESS 0.138, a new option for launching a system with software has been added: **launch from software lists**. This new option is based on the inclusion of software lists, in `.xml` format, containing checksums of known dumps, and it works a bit more like MAME, in the sense that to e.g. launch SNES emulation of Super Mario World requires the following command

```
mess.exe snes -cart smwu
```

where “smwu” must be the proper image which is listed in the software lists “snes.xml”. You can find more details about this new option in the [Software Lists](#) section below.

Reset a System

To reset MESS emulation, press `<key>S-F3</key>`. Notice that, if you are emulating a computer and you are in the fully emulated keyboard mode, `<key>S-F3</key>` will have no Reset effect. In this case, press `<key>ScrLock</key>` (`<key>Ins</key>` for (SDL)MESS) to switch your keyboard in partial emulation mode and press again `<key>S-F3</key>` to actually reset MESS emulation. Notice that this kind of reset function

corresponds to quit and restart completely the emulation. Single systems may offer a different Reset functionality that could be available by pressing a specific key (check [sysinfo.dat](#) to find out if such a functionality is present or not in the system you are emulating)

Change System

To change the emulated system, you can simply quit MESS (by pressing <key>Esc</key>) and start the emulation of a new system. Alternatively, you can enter the Internal UI (by pressing <key>Tab</key>) and choose the Select New System option.

I Found an Issue!

If you find a bug in MESS, please follow these simple steps

1. Check if, at start, you are prompted with a red screen stating the emulation is incomplete and known to be not working, or with a yellow screen listing particular emulation issues. For these systems, problems should be reported only if they crash or freeze MESS. Otherwise, they are probably already known and due to missing feature in the emulation.
2. Check [MAME Testers](http://mametesters.org/) [http://mametesters.org/] to determine if the bug it has already been reported. If not, please register and report the issue you have found.

Also, you may follow the next steps as well

- Try to run MESS with the -log option and check the resulting error.log file with a text editor (e.g. Notepad, Emacs, TextEdit). This output file could contain additional information about the problem (e.g. if you're trying to run a cart format which is unsupported) and should be added to a bug report, if possible.
- If the problem is a crash, and you can compile the emulator by your own, then you might help us by compiling MESS with symbols enabled, i.e. with SYMBOLS=1 (see the [compiling section](#) of this manual), and by adding with your report a backtrace of the crash; such a backtrace can be obtained with “gdb” by following these steps:
 - open a command prompt / bash / terminal and move to the folder where MESS is
 - launch `gdb mess.exe`
 - type `run -window -other_necessary_options` (where “-other_necessary_options” is any other parameter you need to reproduce the bug, e.g. the system, the media and the software names)
 - play until the crash happens
 - type `bt` or `bt full` to obtain a backtrace of the crash
- Alternatively, don't hesitate to ask for suggestions on the [MESS message board](http://forums.bannister.org/ubbthreads.php?ubb=postlist&Board=1) [http://forums.bannister.org/ubbthreads.php?ubb=postlist&Board=1]

If you can provide fixes or addition of missing components, please follow the submission rules at the [contact page](http://mess.redump.net/contacts.php) [http://mess.redump.net/contacts.php].

Key Functions

Here is a table with some key function in MESS (you could notice they have the same functions in MAME). If you are emulating a computer in MESS, notice that these keys will have effect only in Keyboard Partial Emulation Mode (see [Emulated Keyboard](#) section).

Key	Function
<key>Scrl Lock</key> (<key>Ins</key> in (SDL)MESS)	Switch MESS to partial keyboard emulation which lets you use keys like <key>Tab</key> to access UI, <key>p</key> to pause, <key>Esc</key> to quit, etc.
<key>S-Scrl Lock</key>	Paste copied text into the emulated computer

Key	Function
<key>Tab</key>	Access <u>M</u> ESS Internal UI
<key>p</key>	Pause the emulation
<key>S-p</key>	Skip one frame forward if paused
<key>Esc</key>	Quit the emulation
<key>F3</key>	Reset the emulated system
<key>S-F3</key>	Performs a “hard reset”, which tears everything down and re-creates it from scratch. This is a more thorough and complete reset than an <key>F3</key> pressure
<key>F6</key>	Toggle cheat mode (if started with “-cheat”)
<key>F7</key>	Load a saved system state from a slot number (Only few systems support Save States at the moment)
<key>S-F7</key>	Save system state to 1 of 10 slots (Only few systems support Save States at the moment)
<key>F8</key>	Decrease frame skip during emulation
<key>F9</key>	Increase frame skip during emulation
<key>F10</key>	Speed Throttle (Makes system overspeed)
<key>F11</key>	Frames Per Second and Frameskip information
<key>F12</key>	Saves image of system screen to snaps directory (the name of the saved snap can be configured in <u>mess.ini</u>)
<key>Ins</key>	Fast forward. While held, runs the game with throttling disabled and with the maximum frameskip

These are the default keys assignments for these functions in MESS. However, you can fully customize them by pressing the <key>Tab</key> key during the emulation and selecting Input (general) entry. (more details in the Internal UI section).

For more information about loading/saving a game and about saving snapshot in MESS, read also the Save section and the Snapshots section of this Manual, respectively.

Tips

To simplify the use of some system, MESS introduces a few features that are not always known to the user:

- The whole Internal UI is completely clickable with the mouse. Start command line MESS with a double click, and you will be prompted with a random list of the systems you can launch. Click on one of them and the emulation will start. Press <key>Tab</key> during the emulation and you will enter again the Internal UI which allows you to configure Inputs, Video and Sound options by simply clicking on them with your mouse (a complete description of the available options can be found below...)
- In the File Manager and in the Software List menu, the navigation through folders filled of hundreds of files can be made a lot faster, if you know a couple of features of the Internal UI! First of all, <key>PgUp</key>, <key>PgDn</key>, <key>Home</key> and <key>End</key> are fully supported and allow for faster scroll of the list of filenames. Moreover, as of MESS 0.142, you can type the beginning of a filename, and the File Manager will scroll until the file with the closest name. Adding the fact that you can click with the mouse of the file to start it, finding the file you want to launch has never been so easy!
- For many calculator and chess computers, MESS provides an artwork which depicts the original aspect of the emulated machine (see also the Artwork section). In the Video options menu of the UI, you can

turn it on or off. When the artwork is used, you can also click with the mouse on the keys that you see to produce the expected effect on the emulated machine rather than using keyboard inputs. If you find a system using an artwork but not working when you click on it, please report it on the [MESS message board](http://forums.bannister.org/ubbthreads.php?ubb=postlist&Board=1) [http://forums.bannister.org/ubbthreads.php?ubb=postlist&Board=1] (this feature is not currently supported in every system with an artwork, but we are working on it).

- When emulating a computer in MESS, you may want to try some BASIC program you have found in internet. However, the process of writing them in the emulated system is usually quite tedious. This is why MESS allows you to directly paste into the emulated system whatever text you copy from a file! Simply select the program / command you want to enter, copy it (e.g. by using <key>C-c</key> on Windows) and paste it in MESS by pressing <key>S-ScrLock</key> (this is the default key sequence, but you can remap it to any key you want entering the Internal UI, choosing Inputs (general) and then User Interface).
- MESS can load zipped BIOSes based on their CRC checksum. Hence, if you have a correct BIOS zipped in the roms/ folder, the names of the single files are not really important. However, if you keep your BIOS unzipped, MESS cannot load the files by CRC and you need the BIOS files to be named in a very precise way. If you keep receiving a missing files error message and you think you have already the required files unzipped in a subfolder of the roms/ folder, check if their names are correct.

Folder & Files

Shortcuts

mess.ini	ini/ Folder	tools/ Folder
roms/ Folder	ctrlr/ Folder	Effects Files
nvram/ Folder & .nv Files	hash/ Folder	Font Files
cfg/ Folder & .cfg Files	memcard/ Folder	
artwork/ Folder	snap/ Folder	

mess.ini

This is probably the most important file when you use MESS, since it contains all the configuration settings which are used by MESS. By changing mess.ini, you can use to configure every aspect of MESS. Many configuration options and the corresponding command line commands are described in separate sections of this User's Manual and you can find a detailed description of configuration each option at the [mess.ini](#) page of this wiki.

If you cannot find mess.ini in your MESS folder, you can easily create a new mess.ini containing default settings by launching from command line

```
mess.exe -cc
```

or

```
mess.exe -createconfig
```

Afterwards, you can always edit this file and the configuration values by opening and editing it with a text editor. The same command can be also handy if you want to reset every option to its default value at once, because the new mess.ini will overwrite the one already present.

It is generally recommended that you modify the writeconfig setting to 1. Keep in mind that this will also save the information what cartridges and floppy disks are inserted, so they are automatically re-mounted on the next start unless you specify something different. If you prefer to start with a clean system each time you should set this value to 0.

writeconfig 1

Save your changes. Then, copy your new mess.ini to the ini folder.

roms/ Folder

As in MAME, this is the default folder for roms. The main difference is that in this case "roms" refers to BIOS roms of the various supported systems. Therefore, to emulate snes in MESS you just need to put its BIOS in a zipfile named snes.zip inside the roms/ folder (more information about BIOS handling in MESS can be found at [here](#)).

On the other hand, software can be put in any directory you want, as long as the whole software location is passed to MESS at launch. E.g. to play the game foo.smc on snes, it is enough to the correct snes.zip into roms/ and launch at command line

```
mess.exe snes -cart "C:\here_are_my_games\foo.smc"
```

nvram/ Folder & .nv Files

.nv files, created in the nvram/ folder by default, represent the content of each 'Non Volatile RAM' device present on the original system (or in its software). A typical example is provided by console carts which were able to store in SRAM the progresses of the player. MESS backups the system nvram in a nvram/<system>.nv file and software nvram in nvram/<system>/<software>.nv files. Next time you launch MESS, it will look in those locations to load the content of saved memory. These files only contains the NVRAM data that would have been saved in the original systems. For other kind of saving you have to refer to [Savestates](#), which are completely unrelated.

cfg/ Folder & .cfg Files

MESS saves in .cfg files, created in the cfg/ folder by default, every change made by the user in controls menus (Input (general) and Input (this System) in the Internal UI) and configurations menus (Dip Switches, Configurations and Categories in the Internal UI). If you experience any problem after changing these settings and/or you want to restore default settings and controls, you just have to delete from cfg/ folder the file <system>.cfg or default.cfg (the latter for changes in the Input (general) menu).

artwork/ Folder

MESS looks into this folder to find artworks: as for MAME artworks, these are overlay files which can make easier or more satisfactory the emulation. Typical examples are images of calculators, pocket computers and chess computers which allow a much more faithful experience than the simple LED display. MESS also supports clickable artworks which are very useful for calculators, pocket computers and chess computers: by clicking on the keys shown in the artwork, you will pass the corresponding inputs to the emulated system!

ini/ Folder

If you had set writeconfig to 1 (as noted in the mess.ini section above), each system will have its own ini file created the first time you run that system. Each ini file is a complete copy of mess.ini, thus you can customise the settings for each individual system.

When you mount an image (cartridge, tape, etc), that information is saved in the system's ini file, to be remembered for next time.

If for some reason you have an unexpected problem with a system, just delete the corresponding ini file, and the defaults will be used next time.

ctrlr/ Folder

In progress

hash/ Folder

This folder contains .hsi files and .xml software lists for MESS.

.hsi files are xml-like files which contain CRC checksums of software image, information about software known issues (e.g. if a file is a bad dump, has been overdumped with garbage data etc.) and/or specific settings used by MESS to handle software for certain systems. Among others, .hsi files help MESS to identify NES mappers and Intellivision banking schemes based on the checksums of the file images. Most of the .hsi files are based on GoodTools information (thanks to Coverings) but, since the files can be read and edited with any text editor, you can create your own .hsi file if you want to.

.xml software lists are used to identify known dumps and to assign them a shortname which can be used to load the software from command line or from Internal UI or GUI.

memcard/ Folder

In progress

snap/ Folder

In this folder, MESS saves the snapshots you take from the emulated system by pressing <key>F12</key>. The saving scheme depends on the -snapname option chosen (see [this entry](#) of the Manual).

tools/ Folder

In progress

Effects Files

MESS emulates the video output of the emulated systems as accurately as possible. Hence, no enhancement filters or effects are implemented.

However, it is possible to overlay the screen with tiled .png files to mimic and re-create scanlines or tv effects. Some ready-to-use effect files can be found at [The Sticky Collection MAMEWorld Forum](http://www.mameworld.info/ubbthreads/postlist.php?Cat=&Board=emuadvice) [http://www.mameworld.info/ubbthreads/postlist.php?Cat=&Board=emuadvice] (these are for MAME, but they work in the same way in MESS). To enable such effect files, you can load them at command line with the option -effect The .png files must be uncompressed and located in the artwork directory (artwork/ by default). You load them by launching, e.g.

```
mess.exe -effect aperture1x2rb
```

which will force MESS to overlay the screen output with a tiled aperture1x2rb.png pattern, if you have aperture1x2rb.png in the artwork folder.

Font Files

In MESS, you can also customize the font of the Internal UI in a very easy way. Simply put a .bdf file in the same directory as MESS (but you can modify the location by using the -fontpath option), rename it to ui.bdf and, next time you start the emulator, the new font set will be used.

Some ready-to-use font sets in .bdf format can be found at [Mr. Do's artwork pages](http://mrdo.mameworld.info/mame_artwork_supp.php) [http://mrdo.mameworld.info/mame_artwork_supp.php].

Command Line Options

Shortcuts

Overview	Performance Options	Emulated RAM Size Options
Creating mess.ini	Frameskipping	Emulated Media Options
Getting Started	Debugging Options	Internal UI Options
ROM Verification Options	System BIOS Options	
Directories Options	Emulated Keyboard Options	

Overview

Sharing its core with MAME, MESS is a very powerful and flexible emulator. This results in a very large number of command line options available, which can easily lead the user to some confusion. However, in general, most options can be set once and then seldom modified: for this reason is important to keep in mind that to each option below (and to the ones described in the [Video Options](#), [Sound Options](#) and [Controller Options](#) sections of the Manual) there correspond a configuration option in mess.ini. Editing a configuration option in [mess.ini](#) will force MESS to change that particular aspect of the emulation every time it is launched, exactly as if you would pass the corresponding command line option at launch. This approach allows you to only remember a couple of options when you are starting emulation, namely the ones that you want to overwrite for a particular system (and also in this case you may want to modify them once for all creating a <system>.ini file, where “system” is the computer or console you want to emulate)

Also, remember that if you are using MESS in conjunction with a frontend, you can set your favorite configuration options through the frontend's graphical interface rather than launching the emulation from the command line. The option in the frontend should have the same name of the Command Line / Configuration ones.

Notice that here we list and discuss only general MESS options, and only the main ones. For Video, Sound and Controller options, please look at the corresponding entries below. For a complete list of command line options, you can see the [usage](#) page of this wiki, while for a list of the corresponding configuration options you can see the [mess.ini](#) page of this wiki

ROM Verification Options

MESS offers the possibility to verify if the romset you are using contains all the files required for the emulation. You can simply launch the following command to test your romset for “system”

```
mess.exe -verifyroms <system>
```

In the case of e.g. a800pal, you must of course use

```
mess.exe -verifyroms a800pal
```

If the romset is exactly the one needed by MESS, you can receive either a

```
romset <system> is good
```

or a

```
romset <system> is best available
```

message. The former indicates that you have all you need to run emulation in MESS; the latter can appear when the known dump has some issue (e.g. it needs to be redumped or it misses some currently undumped components). In this case, emulation may work or not, depending on the importance of the missing components. However, it also means that no one has a better dump, so you cannot do much to improve your romset.

On the other hand, if the whole romset is missing you will receive a

```
romset "<system>" not found!
```

message

Finally, if your romset is not complete or it contains wrong ROMs, you will receive a

```
romset <system> is bad
```

message, with indication of which files are missing and/or wrong.

If your romset is not good, the only thing you can do is to look for a good one. Also, notice that, if “system” is a clone, i.e. if it could share some ROMs with another system, as in the a800pal case above, you would receive a message like

```
romset a800pal [a800] is best available
```

where the [a800] part means that MESS has also looked for the roms in the parent romset, if present.

If you don't specify any system after the -verifyroms option, MESS will verify the romsets for every supported system.

Directories Options

Basically every directory settings can be customized in MESS, both in mess.ini and at the command line. Namely, you can modify the default locations in which MESS looks for needed files (BIOS ROMs, hash files, artworks etc.) and the default directories in which MESS stores outputs files (cfg, nvram, inp, snapshots etc.).

The former ones can be modified by using the following options, recalling that you have to enter the absolute path for each directory:

```
-rompath [for BIOS ROMs]
-hashpath [for hash files]
-artpath [for artworks]
-ctrlrpath [for controller files]
-inipath [for alternative ini files]
-fontpath [to change fonts in the internal UI]
-cheatpath [for cheat XML files]
```

To changes the output directories from the default ones, you can use the following options, followed by the absolute path of the directory you want to set:

```
-cfg_directory [for configuration files]
-nvram_directory [for save RAM files]
-memcard_directory [for memory card files]
-input_directory [for input recordings]
-state_directory [for save states]
-snapshot_directory [for screenshots]
-diff_directory [for CHD diff files]
-comment_directory [for debugger comments]
```

Check also [this page](#) and [this page](#), respectively, to read about the corresponding configuration options available in mess.ini.

Performance Options

MESS also offers a few options to improve performances.

-sleep

With this option (turned ON by default), MESS gives back as much CPU time as possible to the operating system, when nothing is going on. However, you may want to switch it OFF on some laptops, to improve performances.

-speed

With this option you can force MESS to run a system faster than its normal speed. E.g.

```
mess.exe kaypro2 -speed 1.5
```

will force MESS to run Kaypro 2 emulation at 1.5 times its normal speed.

-refreshspeed

With this option MESS will match its speed to user's monitor refresh rate, to prevent sound hiccups and to ensure a smoother scrolling. However, some adjustment may be still needed by playing a bit with this and the -speed option.

-throttle

With this option, MESS will try to keep the emulated system running in real time. If you switch it off (with -nothrottle), the game will run as fast as your CPU specs allows, ignoring CPU speed and screen refresh of the emulated machines.

-str

With this option, MESS will run for a given amount of seconds and then will exit emulation. E.g.

```
mess.exe a500n -str 20
```

will run Commodore Amiga 500 (NTSC) emulation for 20 seconds and then quit. This option can be of use for benchmarking and regression testing.

Frameskipping

If an emulated system cannot run 100% on your system, you might want to sacrifice some frame here and there to avoid slowdowns. This can be obtained by using -autoframeskip. MESS will try to keep the speed of the game as closer as possible to the intended one, automatically choosing how many frames to drop.

Alternatively, you can tell MESS how many frames out of 12 to drop, by setting -frameskip n at launch, with n an integer number between 0 (default) and 12. You can also change this setting during emulation by pressing <key>F8</key> and <key>F9</key> (see the [Key Functions](#) section).

Keep in mind that dropping too many frames will cause the game to become more jerky and stuttery.

Check also [this page](#), to read about the corresponding configuration options available in mess.ini.

Debugging Options

-log

When running MESS with `-log` option, the emulator will generate a file called `error.log` which contains error information and debug messages. It can help to track the reason of a bug.

-debug

When running MESS with `-debug` option, the emulator will start with the debug window active. From there, you can disassemble CPU instructions, follow memory accesses, read error messages that would be added to `error.log` and look for memory locations to be modified for cheats. To learn more about the debugger, simply type `help` in the debugger window, or read this [wiki page](#) devoted to the topic.

System BIOS Options

Many systems in MESS supports multiple bios choices (e.g. bugfixed and/or improved versions of the OS). These can be chosen at command line, by using the `-bios` option. E.g. `nascom1` supports versions T1, T2 and T4 of its operating system NasBug. You can select which one you want by launching MESS with

```
mess.exe nascom1 -bios T1
mess.exe nascom1 -bios T2
mess.exe nascom1 -bios T4
```

respectively. Please refer to [sysinfo.dat](#) for the available BIOS options for each system.

Emulated Keyboard Options

For emulated computers and calculators, MESS offers two different way to translate the keyboard of the original system into the keyboard of your computer:

- the **emulated** mode, in which original keys are mapped to your keyboard based on their position (e.g. `<key>Z</key>` could be mapped on your `<key>Y</key>` if you're emulating a QWERTZ keyboard with your QWERTY one): this can sound a bit unnatural, but it can be handy when you're looking for an unusual key which was only used by home computers in the '80s and has no correspondent key on modern keyboard (e.g. Atari or Commodore function keys on 8 bit home computers); also you can remap every single key by pressing `<key>Tab</key>` and entering the `Input (this System)` menu of the Internal UI
- the **natural** mode, in which original keys are mapped to the equivalent keys of your keyboard when present (e.g. your `<key>Y</key>` always produces a 'Y', no matter if the keyboard in the emulated system was QWERTY or QWERTZ); this is more natural to be used but some keys are lost in the translation, e.g. again the old Atari or Commodore function keys have no equivalent on modern keyboards.

By default, you start MESS in emulated mode. However, you can start emulation in natural mode by using the `-natural` option at command line or by changing the corresponding configuration option in `mess.ini`.

Also, remember that you can switch keyboard mode in any moment, by pressing `<key>Tab</key>` and entering the `Keyboard Mode` menu. This allows to easily remap and use missing keys when you need them. See the [Emulated Keyboard](#) section below for more details.

Emulated RAM Size Options

MESS offers different RAM sizes for many emulated computers, reflecting the possible configurations of the original systems. You can select the RAM size by using the `-ramsize` option. Refer to [sysinfo.dat](#) to find which valid sizes are available for each system.

Emulated Media Options

MESS supports for each system a certain number of media devices (tapes, floppy disks, cartridges) and file formats. A list of the available media and the correspondent supported file formats can be obtained at command line by using `-listmedia` which lists the available media of a given system, e.g.

```
mess.exe <system> -listmedia
```

If you don't specify any system, i.e. if you simply use

```
mess.exe -listmedia
```

you will obtain a list of each supported system with its corresponding media devices and file formats.

While usually media are named in a standard way (e.g. `flop1` for the first disk drive, `flop2` for the second one, `cass1` for the first tape drive, etc.), some systems require specific names to work. Hence, you shall remember to use the `-listmedia` option if you get an “unknown option” error message. You also may check [sysinfo.dat](#) to read a list of supported devices and file formats for each system.

Software Lists

Shortcuts

Overview	Usage from Command Line	Additional Usage
List Format	Usage from Internal UI	Using Software Lists with cmpro
Setting Up Lists		Software Sources
Setting Up Software		Missing Features

Overview

As of 0.138 release, MESS has added support for software lists in xml format to document the games & programs which have been released for various consoles and home computers, much like the source already does for the systems themselves and MAME does for the arcade games. At the moment, not all the systems are covered, but the plan is to extend the support to most systems, given the high risk for info on obscure and not so obscure systems & software to be lost forever.

The software lists are stored in the `hash/` directory and are released together with the emulation. Hence, you should have them already both if you download the current MESS release and if you downloaded the latest svn. Alternatively, you can download the various xml lists (or check their content) from our [git repository](http://git.redump.net/mess/plain/hash/) [<http://git.redump.net/mess/plain/hash/>].

Notice that software lists are an additional documentation effort of MESS, but they are not going to completely replace fullpath loading. Indeed, we are fully aware that part of the fun when emulating an home computer is the capability for the user to run its own software, and therefore loading by fullpath will always remain available as well.

Note: by mistake, in version 0.142 the lists have been only included in the source pack and not in the exe pack, please get them from the git repository until next version is released

List Format

We have chosen a (minimal) format which contains a **shortname** for each game (like MAME shortnames), a **description** (which should match the title screen, or the box screen if no title screen exists), the **release year**,

the **publisher name** and the **CRC32/SHA1 checksums**, plus some additional fields that helps either the emulation or the accurate description of the original hardware. Additional info strictly concerning the software, like available languages, programmers names, cheats, etc. can be stored externally in datfiles to be displayed by front-ends.

More info about our software list format can be found in the [appropriate Wiki page](#).

Setting Up Lists

By default, software lists in .xml format are stored in the hash/ directory inside the MESS directory. You can however decide to store them elsewhere (or in multiple paths), by adding the corresponding directories with the hashpath option.

Setting Up Software

Software storage is slightly more complicate than list storage. The basic principles is that software for each list should be stored in a directory named after the list name and placed inside your roms/ directory. E.g. the Epoch Super Cassette Vision carts, which belong to the scv.xml list, should be stored in a roms/scv/ directory and the Sharp X68000 floppies, which belong to the x68k_flop.xml list, should be stored in a roms/x68k_flop/ directory.

In this way, you can separately store the software of different systems and the different media of each system (e.g. carts separated from disks separated from tapes).

Of course, MESS also allows you to store your carts in different folders than “roms”, e.g. you can use “C:\console_games\snes” and “C:\console_games\gbcolor”, as long as you add the root directory “C:\console_games” to your rompaths in mess.ini. The only things you have to care about is that the subfolders (“snes” and “gbcolor”, in the examples above) are correctly named as the software lists they refer to.

Notice that, for helping development, MESS also looks for list items directly in the roms/ directory. Hence, you can in principle store all your software in the rompath, together with the bios files, even if it is not the recommended setup.

Usage from Command Line

If you have stored the software roms as explained above, and you want to launch “Super Mario Land” for the Game Boy (which corresponds to the shortname sm1 in gameboy.xml), then you simply have to type

```
mess gameboy -cart sm1
```

and MESS will load the game.

Notice that if you type the wrong shortname, then a few best match suggestions will be prompted to you, so that you can amend the game name.

And if you want to launch “Ancient Ys Vanished Omen” for the Fujitsu FM7 (which corresponds to the shortname ys in fm7_disk.xml), you can simply type

```
mess fm7 -flop1 ys
```

or

```
mess fm7 -flop2 ys
```

depending on which disk drive you want to use. In fact, you can also use

```
mess fm7 -flop1 ys -flop2 ys
```

but in this case you end up mount the first disk of the game in both disk drives.

If you want to mount different disks in the various drives, you can do that as well! To load e.g. the third disk of the game in the second drive, you have to use the following command

```
mess fm7 -flop1 ys -flop2 ys:flop3
```

where the `:flop3` part tells **MESS** to load the data corresponding to the “flop3” part in the xml file. If you specify a part which does not exist (e.g. because of a typo or because you think a game has more disks than there really are), **MESS** will revert to loading the first disk in the drive and you will have to change disk during emulation, through the File Manager in the Internal UI.

Also, notice that some systems in **MESS** support not only their own lists but also some compatible ones. For example, **gbcolor** supports `gameboy.xml` as compatible list, in addition to its own `gbcolor.xml` list of carts. This means that you can load any gameboy game in **gbcolor**: using the same “Super Mario Land” example above, you can launch `sm1` in the Game Boy Color, by using the following command

```
mess gbcolor -cart gameboy:sm1
```

where the `gameboy:` part tells **MESS** that `sm1` belongs to the `gameboy.xml` list. Similarly, you can load “Legend of Zelda Oracle of Ages” for the Game Boy Color (whose shortname is `zeldaage`) both in **gbcolor** with the command

```
mess gbcolor -cart zeldaage
```

and in **gameboy** with the command

```
mess gameboy -cart gbcolor:zeldaage
```

even in this specific case you will receive a “This Game Can Only Be Played on a Game Boy Color” message, as if you had plugged your “Oracle of Ages” cart inside an old Game Boy console.

Other examples of compatible lists are `wscolor.xml` for the Wonderswan, `wswan.xml` for Wonderswan Color, and `pce.xml` for SuperGrafx.

Usage from Internal UI

In addition to support from command line, software lists are completely supported in the File Manager of the Internal UI. Once emulation is running, indeed, you can press `<key>Tab</key>` and select the **File Manager**. Once you have chosen a media device that you want to fill with an image, the File Manager allows you to browse your computer in search of the files to mount. Among available options, the third from top is the option you are looking for: **Software List**!

Once you have chosen it, you will be prompted with the available lists for the system you are running and the specific media device you had selected in the File Manager, including any available compatible list. In the examples above, when trying to load a FM7 floppy disk you would find here the “Fujitsu FM-7 Disk Images” and when trying to load a Game Boy cart you would find here both the “Nintendo Game Boy cartridges” and the “Nintendo Game Boy Color cartridges”.

Select the list you want, and you will access the menu with all the games present in our xml software list. Launching the game you want is now a matter of a click! And if you select a game which is composed of multiple media (e.g. multiple cdroms or multiple floppy disks), you will be prompted with a new submenu which allows you to select among the available parts the one you want to load.

Notice that by default games are ordered based on their shortname, but you can select the first option at top to display the list of games ordered by their description.

Tips - In the **Software List** menu, the navigation can be made a lot faster if you know a couple of features of the Internal UI! First of all, `<key>PgUp</key>`, `<key>PgDn</key>`, `<key>Home</key>` and

<key>End</key> are fully supported and allow for faster scroll of the list of filenames. Moreover, as of MESS 0.142, you can type the beginning of a filename, and the menu will scroll until the file with the closest name (which will be the closest shortname or the closest description, depending on the ordering you have chosen). Adding the fact that you can click with the mouse of the file to start it, finding the file you want to launch has never been so easy!

Note This specific entry of the Manual refers to latest svn builds. Some features have been indeed added after 0.142 got released.

Additional Usage

One of the main reasons to have a database of known dumps, is to be able to identify spare files or dumps you do by yourself. This has been an appreciated feature in MAME for years, when dealing with arcade dumps, and we wanted to offer something similar for consoles and home computers. Software lists allow for this: indeed, if you have some unknown files in “C:\unk\files\”, then you can simply launch

```
mess -romident C:\unk\files\
```

and MESS will compare size and checksums with the ones recognized by its source (for BIOS items) and by its software lists (for cart/floppy/tape items).

Using Software Lists with cmpro

Software lists are not 100% integrated with clrmame, hence not everything is automated, but with some work at your end you can successfully use MESS xml lists with cmpro.

Single list handling

First of all, let assume you are only interested in a single software list, say the Arcadia one. Since at the moment clrmame cannot directly extract the list info from MESS, you have to manually feed the software list into the rommanager, by pointing cmpro directly to the arcadia.xml file in the “hash/” directory (by selecting “Add DatFile...” in the Profiler window, and browsing your HD until you find the “hash/” directory inside your MESS directory). Once you have added the list to clrmame, a new profile will be created and, by loading it, you can finally setup the directory where you want to store your arcadia software, and rebuild any directory containing arcadia roms to create a romset compatible with MESS.

Multiple lists handling

Now, what if you want to get all the supported lists at once? In this case, a first dumb solution would be to repeat for each list the procedure above. However, there is also a smarter and easier procedure, thanks to the clrmame batchrun capabilities! Indeed, to rebuild all the software lists at once, you can simply follow these steps:

1. you point clrmame to all the lists you are interested in (by selecting “Add DatFile...” in the Profiler window and choosing the xml lists you want)
2. after clrmame has created a new profile for each of the lists, you select all of them
3. by pressing Enter, you are prompted with a “Profile Batchrun” window where you can choose among plenty of batchrun options
4. in the Misc submenu, you select the fundamental “Create rompath for new dat” option and choose as root folder either your roms/ directory or any other rompath you like (you can leave “For rompath naming use” to its default value “Datfile Name Tag”)
5. in the Rebuilder submenu, you select “Rebuilder run before scan” and “Use source folder”, and you enter the location of the directory which contains the roms you want to rebuild
6. you press OK and you wait for clrmame to do its job (remember to confirm OK when the statistics are prompted)

At the end, you will have the folders you need inside “roms/” and all the available roms will be in the correct location. Of course, at Step 5. you can also select additional Rebuild and Scan options as you like, or you can decide to only Rebuild without scanning.

Notice that, after the first run, you can always select multiple profiles to scan more software lists at once, with or without any Rebuild procedure.

Conclusion

Eventually, we hope to add better integration between clrmame and MESS software lists (e.g. clrmame could be able to choose proper batchrun options after simply parsing the MESS -listxml output), but for the time being the procedure described above should work for everyone with little effort.

Software Sources

First of all, you should remember that software which have been released and sold, no matter how old, is still copyrighted material. Therefore, unless the original owner of the rights has released it for public use, it is illegal to download such a software.

In some countries, it is legal to make a backup copy of a software if you own it. In this case, you can indeed copy your software and use it with our lists. However, in many countries, even backup copies of the software you own is a sort of gray area.

As a result, we do not distribute any software with MESS and it is completely up to the user to look for it.

What we distribute is a series of files containing info about software titles and their checksums. Since our goal is to document **accurately** the original software, it is however of interest to explain where we got these info from.

Things change a lot depending on the media we are talking about and as such we cover them separately here.

Cartridges

For carts, we have either relied directly on dumper's info or on collections like the no-intro ones, which currently represent the most up-to-date collection concerning carts for a lot of systems.

For pirate carts, info come usually or from alternative collections (like GoodTools or TOSEC), or from chinese dumpers and chinese message board members.

Overall, when you want to verify if the dump you have matches one of our software lists, you have to remember that we try to match the original cart contents as much as possible. Therefore, cart dumps in our lists do not have headers and consist on separate files if it is known that the cart contained more than one chip.

If you have a file in a different format, you can often recover the files with the expected checksums by following the following steps:

1. open the file with an hex editor, and strip the header (whose size depends on the file format: e.g. the iNES and A78 header is typically 0x10 long).
2. if you have a single file, while the list expect multiple ones, you can try to split the file you have according to the size expected by the list, and check if the resulting part gives you the right files.

Point 1. could help for NES and A7800 mainly (not many other headered format are around), point 2. helps for sure with NES (separate PRG and CHR), with Coleco (most carts contained 8K rom chips only) and with latest additions to the SNES list.

Note - Among current lists, lynx.xml still contains files with header. This is going to be rectified soon, anyway.

CDs

For CDs, as for carts, we try to rely on the best collections available, which often means redump.org, but not always (e.g. they still haven't found a way to extract properly the content of PCE CDs). Then, we convert the available .bin + .cue files to .chd, following the instructions that you can find [here](http://guru.mameworld.info/tutorials/dumping_cds_&_hdds/index.html) [http://guru.mameworld.info/tutorials/dumping_cds_&_hdds/index.html].

This is usually enough to produce the .chd with the checksums expected by the software list.

Things become a bit messier when the only available rips are in .iso+.mp3 format. In this case, the conversion to .chd usually involve in mounting the image in a program like Daemon Tools and then ripping it as a real CD.

Floppy Disks & Tapes

For floppies and tapes the situation is not yet established. We currently use common formats like .imd or .td0 or .d88 (for disks), and .t77 or .tap (for tapes, which got converted internally to .wav at loading time), but we might decide in the future to witch to some better format if a good candidate appears.

Missing Features

There are various aspects which still need to be improved.

Core improvements

As already mentioned, we need to complete integration of lists with the UI, to both display software info of the selected image and to handle software consisting of several files.

List improvements

While lists already contain most of the known software for each systems, additional info like year and publisher are often incomplete or missing. And even when present, they often still have to be verified against title screens.

Console Emulation

Shortcuts

Available Emulated Devices, Formats and Media	Using Memory Cards	Emulated Controllers (Consoles)
---	------------------------------------	---

Available Emulated Devices, Formats and Media

For each emulated system, MESS supports a certain number of media devices present on the original system, e.g. carts or CD-ROM players, and a certain number of file extensions for each device. To find out which devices and formats are supported by your favorite system, simply launch

```
mess.exe -listmedia <system>
```

Assuming you were interested in a2600, you will obtain the following output

SYSTEM	DEVICE NAME (brief)	IMAGE FILE EXTENSIONS	SUPPORTED
a2600	cartridge (cart)	.bin	.a26
	cassette (cass)	.wav	.a26

Device name and its brief version can be used at command line to launch a specific program/game. E.g.

```
mess.exe a2600 -cartridge foo1.bin
```

or

```
mess.exe a2600 -cass foo2.a26
```

Notice that only supported extensions will load successfully.

In the following, we briefly describe how to use media in console emulation under MESS. See also the Media Emulation section of this Manual.

Notice that, starting from MESS 0.138, it is also possible to load images from software lists: instructions are the same as the ones described below (except you have to use the appropriate name for the software, as prescribed in the .xml list). See the Software Lists section of this Manual for all the details.

Loading a Cartridge

You can load a cartridge in MESS it by simply launching

```
mess.exe <system> -cart foo.ext
```

If a system supports more than one cartslot, you may need to replace -cart with either -cart1 or -cart2.

Alternatively, while the emulation is running, you can enter the Internal UI, choose the File Manager menu, select one of the Cartslot entries and browse your HD to find the cart file you want. After selecting it, the emulation will be automatically restarted with the cart inserted.

Notice that a few systems currently supports 'fake' multiple slots to allow the emulation of carts with multiple files. This will be fixed eventually with the introduction of a xml format for multi-file cartridge, but for the moment you may need to use -cart1 & -cart2 also in this case.

Loading a CD-ROM

For systems which support CD-ROM emulation, MESS uses CHD file format. You can read a guide about how to create a CHD from a CD [here](http://guru.mameworld.info/tutorials/dumping_cds_&_hdds/index.html) [http://guru.mameworld.info/tutorials/dumping_cds_&_hdds/index.html] , courtesy of The Guru. Once you have a CD-ROM image in CHD file, you can load it by simply launching

```
mess.exe <system> -cdrom foo.chd
```

Alternatively, while the emulation is running, you can enter the Internal UI, choose the File Manager menu, select Cdrom and browse your HD to find the CHD file you want. After selecting it, the emulation will be automatically restarted with the CD inserted.

Loading a Cassette / Tape

You can load a tape in MESS it by simply launching

```
mess.exe <system> -cass foo.ext
```

If a system supports more than one cassette drive, you may need to replace -cass with either -cass1 or -cass2.

Alternatively, while the emulation is running, you can enter the Internal UI, choose the File Manager menu, select one of the Cassette entries and browse your HD to find the tape file you want. After selecting it, the emulation will be automatically restarted with the cassette inserted. Depending on the system, you may need to manually press “Play” to start cassette loading (or, as in the c64 case, to type “LOAD” and to press “Play” afterward). To do this in MESS, simply enter the Internal UI, choose the Tape Controls menu and press “Play”. The same procedure goes for any other tape operation (“Rec”, “Stop”, etc.).

Notice that the loading procedure can be slow. However, you can speed up emulation with no bad consequence by pressing <key>Ins</key> in partial keyboard emulation mode.

Loading Quickloads

For some systems MESS also supports quickload files. These are usually dumps of the system memory which can be re-inserted in the emulated machine to make a program or a game start immediately. To load a quickload file in MESS, you can simply launch

```
mess.exe <system> -quick foo.ext
```

Alternatively, while the emulation is running, you can enter the Internal UI, choose the File Manager menu, select Quickload and browse your HD to find the quickload file you want. After selecting it, the emulation will be automatically restarted with the quickload file loaded into memory.

Removing a Cartridge

While the emulation is running, you can enter the Internal UI, choose the File Manager menu, select Cartslot and click on the cart file you are currently using. This shall prompt you with a file browser which allows you either to change cart image or to select “empty slot” to remove completely the cart file from the cartslot.

Alternatively, you can quit emulation and start it again with a new cart image.

Removing a CD-ROM

While the emulation is running, you can enter the Internal UI, choose the File Manager menu, select Cdrom and click on the CHD file you are currently using. This shall prompt you with a file browser which allows you either to change CD image or to select “empty slot” to remove the CHD file from the CD-ROM player.

Alternatively, you can quit emulation and start it again with a new CD-ROM image.

Removing a Cassette / Tape

While the emulation is running, you can enter the Internal UI, choose the File Manager menu, select Cassette and click on the tape file you are currently using. This shall prompt you with a file browser which allows you either to change tape image or to select “empty slot” to remove completely the tape file from the cassette drive.

Alternatively, you can quit emulation and start it again with a new tape image.

Removing Quickloads

Quickloads will remove themselves after use. You don't need to do anything.

Saving to a Cassette

For systems supporting cassettes, you can save programs to tape exactly as in the original machines. First of all, you need to have a cassette in the tape drive: enter the Internal UI in MESS, choose the File Manager

menu and select Create to create a new cassette file (in .wav format) with any name of your choice. Now, you will see that a brand new tape file is inserted in the Cassette slot. Next step consists in using the procedure that you would have followed on the real machine to save on tape (e.g. for c64 you have to type SAVE after having written the program). Once the emulated computer is ready to save, you just have to press “Rec” on the cassette drive. In MESS, this means to enter the Internal UI, select the Tape Controls menu and press “Rec”. At this point, MESS will save the output to the .wav file you created. Next time you use MESS, you can also load the program again, exactly like a cassette on the emulated system!

Notice that the saving procedure can be slow. However, you can speed up emulation with no bad consequence by pressing <key>Ins</key> in partial keyboard emulation mode.

Using Memory Cards

In progress

Emulated Controllers (Consoles)

Emulated Joystick

For consoles using joystick/joypad controllers, MESS follows usually MAME default keyboard mappings:

Arrows Key	Directions
<key>1</key>	Start
<key>5</key>	Select
<key>LCtrl</key>	Button1
<key>LAlt</key>	Button 2
<key>Space</key>	Button3
<key>LShift</key>	Button4
<key>Z</key>	Button5
<key>X</key>	Button6
<key>c</key>	Button7
<key>V</key>	Button8

Also, if you enable your joystick with -joystick at command line (or in mess.ini or in GUI), you will usually have the emulated joystick automatically mapped to your joystick. However, you can re-map each key to the keyboard or to your joystick by entering Internal UI (pressing <key>Tab</key>) and choosing Input (general) and Input (this System) menus.

Emulated Lightgun

For consoles using lightgun controllers, MESS follows usually MAME default keyboard mappings:

Arrows Key	Directions
<key>LCtrl</key>	Button1
<key>LAlt</key>	Button 2

Also, if you enable your mouse with -mouse at command line (or in mess.ini or in GUI), you will usually have the emulated lightgun automatically mapped to your mouse. However, you can re-map each axis / key

to the keyboard or to your joystick by entering Internal UI (pressing <key>Tab</key>) and choosing Input (general) and Input (this System) menus.

Notice that, to enable emulated lightguns as controllers for a console, you may need to enter the Internal UI (pressing <key>Tab</key>) and choose the Categories menu: there, you can usually select which kind of controller you want to connect to each controller port of the emulated machine (this ideally corresponds to the act of connecting the controller on the real thing). You can check [sysinfo.dat](#) to find if the system you want to emulate currently supports emulated lightguns.

Custom Controllers

Some consoles may use custom controllers. These are usually enabled by entering the Internal UI (pressing <key>Tab</key>) and choosing the Categories menu, where you can decide which controller to use among the ones emulated by MESS. After having enabled the controller, you can enter the Input (this System) menu to find out what is the default mapping of the chosen controller and to remap it as it best suites your needs.

You can check [sysinfo.dat](#) to find out which custom controllers are supported for each system.

Computer Emulation

Shortcuts

Pasting Code to Emulated Computers	Available Emulated Devices, Formats and Media	Emulated RAM	Emulated Keyboard	Other Emulated Controllers (Computers)
--	---	------------------------------	-----------------------------------	--

Pasting Code to Emulated Computers

One of the main interesting aspects, when you emulate old home computers, is the possibility to run on them programs you have written, or programs whose code is available in the net. However, having to type hundreds of lines of code into the emulated terminal is not so funny. This is one of the reasons you may love the pasting feature MESS offers. Indeed, with MESS, you can simply copy any kind of code from a webpage or a text file and then directly pasting it into the emulated computer. Once you have copied the text (e.g. with <key>C-c</key> in Windows), you simply have to start the emulation and press <key>S-ScrLock</key> (but, as usual, you can remap this input in the Input (general) menu of the Internal UI).

Available Emulated Devices, Formats and Media

For each emulated system, MESS supports a certain number of media devices present on the original system, e.g. carts or CD-ROM players, and a certain number of file extensions for each device. To find out which devices and formats are supported by your favorite system, simply launch

```
mess.exe -listmedia <system>
```

Assuming you were interested in atom, you will obtain the following output

SYSTEM	DEVICE NAME (brief)	IMAGE FILE EXTENSIONS SUPPORTED
atom	quickload (quik)	.atm
	cassette (cass)	.wav
	printer (prin)	.prn
	floppydisk1 (flop1)	.ssd
	floppydisk2 (flop2)	.ssd

Device name and its brief version can be used at command line to launch a specific program/game. E.g.

```
mess.exe atom -flop1 foo1.ssd
```

or

```
mess.exe atom -cassette foo2.wav
```

Notice that only supported extensions will load successfully.

In the following, we briefly describe how to use devices in computer emulation under MESS. See also the Media Emulation section of this Manual.

For cartridges, cassette tapes and quickloads, see the corresponding section under console, [here](#).

Saving to a Floppy Disk

In Progress

Formatting and Creating a Floppy

In Progress

Emulated Hard Disks

In Progress

Formatting and Creating an Hard Disk

In Progress

Emulated Printers

In Progress

Emulated Serial Ports

In Progress

Emulated RAM

MESS often supports different RAM configurations for a single computer. To choose the configuration you want, you can use the `-ramsize` option at launch. A list of available ram sizes for each system is available in [sysinfo.dat](#).

Emulated Keyboard

Dealing with the emulation of a wide range of systems, from console and handhelds devices, to home computer, calculators and chess computers, MESS has to emulate very different kinds of controllers. In particular, it has to emulate the original keyboard of a system through the keyboard of your computer. However, due to differences in the keyboard layouts (e.g. QWERTY vs. QWERTZ vs. AZERTY vs. Japanese vs. Russian and Bulgarian keyboards) and the possible presence of custom keys in the emulated system (e.g. the 'Apple' command key in Macintosh, the 'CBM' key in Commodore computers etc.), it is tough to cover all the possibilities and to offer at the same time a nice interface for configurations. MESS solution to this problem is to offer two different emulation modes for the keyboard of each system, Full and Partial, whenever you start the emulation of a computer (this does not apply to console). As said above, you

can switch between the two using `<key>ScrLock</key>` (in (SDL)MESS, the default key is `<key>Delete</key>` and you can change it in [mess.ini](#) changing the `uimodekey` line).

(For more information on how to change this `<key>ScrLock</key>` to be something else, see: [for some reason esc does not quit mess nor does tab bring up the menu](#)).

Below is a short description of the differences between the two modes.

Partial Emulation Mode

In this emulation mode, you can access the UI keys described [above](#) (e.g. `<key>Tab</key>`, `<key>Esc</key>`, `<key>F4</key>`, `<key>F12</key>`, etc.). Therefore, this suits your needs whenever you want to configure Inputs and Dip Switches, to take Screenshots, Movies or Savestates, or simply to quit emulation. Notice that, anyway, inputs are registered also by the emulated system, so if you e.g. pause the emulation with `<key>P</key>` in Partial Emulation, a 'P' will appear on the emulated screen as well.

Full Emulation Mode

In this emulation mode, the UI keys effect vanishes and each key on your keyboard corresponds to a key on the keyboard of the emulated computer. Actually, MESS offers two different flavours of Full Emulation: **emulated** and **natural**. You can switch between pressing `<key>Tab</key>` and entering the Keyboard Mode menu.

Full Emulation Mode with Emulated Keyboard (default behaviour)

By default MESS in Full Emulation mode tries to change your keyboard into the original keyboard. This way, all accessible keys of the original keyboard have a correspondent mappings on your keyboard, even the strangest function key that could have been present. Moreover, the default mapping tries to be as close as possible to the original one so that, if you were used to type without even looking at the keyboard on the real thing, you can do the same in MESS! However, this can produce a few strange situations when the original keyboard layout was very different from modern keyboards layout. As an example, you can think to Mattel Aquarius. In this computer the `<key>Return</key>` was at the right of the `<key>. ></key>`, and therefore it defaults in MESS to your `<key>/ ?</key>` key.

Even with this mapping quirks, sometimes it is still not really possible to remap the keyboard of an old computer to a modern keyboard. Hence, some of the following exceptions can happen:

- 'Spacebar', 'Backspace' and the arrow keys (when they have separate keys) are usually mapped to the correspondent keys of your keyboard;
- when an emulated keyboard presents keys that are not on a modern pc keyboards, then MESS tries to map them to a closer key
- character keys at left of '1' or between 'Shift' and 'Z' are often mapped to '~',
- a third character key after 'O' and 'P' or a fourth one after 'L' and 'M' is often mapped to 'Backlash2' or to 'Page Up' and 'Page Down' etc.

To be sure of the mapping of a keyboard in emulated mode, you can verify it in the Input (this System) menu of the Internal UI. In this menu, you can also remap every key to the location you prefer on your keyboard.

Full Emulation Mode with Natural Keyboard

This mode can be activated launching MESS with the `-natural` option or by entering the Internal UI and choosing the Keyboard Mode menu. When you select the Natural Keyboard mode, MESS translates the key you press in the same exact key of the original keyboard, so that when you press e.g. `<key>+</key>` you will see on screen a plus sign, no matter where that key was located on the original system, or if a key combination was needed to access it. Or, again, if you press arrow keys while emulating a Spectrum 48k, you will see the cursor move in the corresponding directions even if you should have used `<key>S-5</key>`,

<key>S-6</key>, <key>S-7</key> or <key>S-8</key> to obtain the same result on a real Spectrum keyboard. This makes typing much easier but it presents a couple of limitations:

- many custom key have no equivalent on modern pc keyboards: in this case, you may want to switch between Natural and Emulated mode when you need such keys
- some UTF-8 characters are not present on every keyboard (e.g. £, ¥, §, π etc.): in this case, usually MESS maps them to the corresponding keys and if you don't have them on your keyboard, you shall switch to Emulated mode when you need it.

Notice that the Natural Keyboard cannot be remapped.

Summing Up

A few experiments combining the three modes, and a bit of time spent configuring the keys you use most, shall allow you to recreate a perfect experience of the emulated system. However, if you experience any problem, or if you think any important key is still unmapped, please file a bug report in [MAME Testers](http://mametesters.org/) [<http://mametesters.org/>] or contact us at the [MESS message board](http://forums.bannister.org/ubbthreads.php?ubb=postlist&Board=1) [<http://forums.bannister.org/ubbthreads.php?ubb=postlist&Board=1>].

Other Emulated Controllers (Computers)

Emulated Mouse

For computers using a mouse, MESS follows usually MAME default keyboard mappings:

Arrows Key	Directions
<key>LCtrl</key>	Button1
<key>LAlt</key>	Button 2

With some computers however, this would conflict with other keyboard functions. You can enter Internal UI and choose Input (this System) to verify the default mappings for the system you are emulating.

Alternatively, if you enable your mouse with -mouse at command line (or in mess.ini or in GUI), you will usually have the emulated mouse automatically mapped to your mouse. However, you can re-map each axis / key to the keyboard or to your joystick by entering Internal UI (pressing <key>Tab</key>) and choosing Input (general) and Input (this System) menus.

Notice that, to enable emulated mouse as controllers for a computer, you may need to enter the Internal UI (pressing <key>Tab</key>) and choose the Categories menu: there, you can usually select which kind of controller you want to connect to each controller port of the emulated machine (this ideally corresponds to the act of connecting the controller on the real thing). You can check [sysinfo.dat](#) to find if the system you want to emulate currently supports emulated mouse.

Emulated Joystick

For computers using joystick/joypad controllers, MESS follows usually MAME default keyboard mappings:

Arrows Key	Directions
<key>1</key>	Start
<key>5</key>	Select
<key>LCtrl</key>	Button1
<key>LAlt</key>	Button 2

<key>Space</key>	Button3
<key>LShift</key>	Button4
<key>Z</key>	Button5
<key>X</key>	Button6
<key>c</key>	Button7
<key>V</key>	Button8

With some computers however, this would conflict with other keyboard functions. You can enter Internal UI and choose Input (this System) to verify the default mappings for the system you are emulating.

Alternatively, if you enable your joystick with -joystick at command line (or in mess.ini or in GUI), you will usually have the emulated joystick automatically mapped to your joystick. However, you can re-map each key to the keyboard or to your joystick by entering Internal UI (pressing <key>Tab</key>) and choosing Input (general) and Input (this System) menus.

Notice that, to enable emulated joysticks as controllers for a computer, you may need to enter the Internal UI (pressing <key>Tab</key>) and choose the Categories menu: there, you can usually select which kind of controller you want to connect to each controller port of the emulated machine (this ideally corresponds to the act of connecting the controller on the real thing). You can check [sysinfo.dat](#) to find if the system you want to emulate currently supports emulated joysticks.

Emulated Lightgun / Lightpen

For computers using lightgun / lightpen controllers, MESS follows usually MAME default keyboard mappings:

Arrows Key	Directions
<key>LCtrl</key>	Button1
<key>LAlt</key>	Button 2

With some computers however, this would conflict with other keyboard functions. You can enter Internal UI and choose Input (this System) to verify the default mappings for the system you are emulating.

Alternatively, if you enable your mouse with -mouse at command line (or in mess.ini or in GUI), you will usually have the emulated lightgun / lightpen automatically mapped to your mouse. However, you can re-map each axis / key to the keyboard or to your joystick by entering Internal UI (pressing <key>Tab</key>) and choosing Input (general) and Input (this System) menus.

Notice that, to enable emulated lightguns / lightpens as controllers for a computer, you may need to enter the Internal UI (pressing <key>Tab</key>) and choose the Categories menu: there, you can usually select which kind of controller you want to connect to each controller port of the emulated machine (this ideally corresponds to the act of connecting the controller on the real thing). You can check [sysinfo.dat](#) to find if the system you want to emulate currently supports emulated lightguns / lightpens.

Custom Controllers

Some computers may also use custom controllers. These are usually enabled by entering the Internal UI (pressing <key>Tab</key>) and choosing the Categories menu, where you can decide which controller to use among the ones emulated by MESS. After having enabled the controller, you can enter the Input (this System) menu to find out what is the default mapping of the chosen controller and to remap it as it best suites your needs.

You can check [sysinfo.dat](#) to find out which custom controllers are supported for each system.

Chess Computer Emulation

Most of the information about consoles apply to chess computers as well (even if in this case devices and media are not very common). However, if the emulated chess computer comes with an artwork (e.g. mm5), you may find interesting the following feature.

Clickable Artwork

One of the main features of chess computer emulation in MESS is the possibility to click on the artwork to pass inputs to the emulated machine. Try to move your mouse on the MESS window and to click on keys present on the artwork. The inputs shall be recognized by the system you are emulating and produce result as if you were pressing the key with your finger!

Calculator Emulation

Most of the information about computers apply to calculators as well (even if in this case devices and media are not very common). However, if the emulated calculator comes with an artwork (e.g. hp48sx), you may find interesting the following feature.

Clickable Artwork

One of the main feature of calculator emulation in MESS is the possibility to click on the artwork to pass inputs to the emulated machine. Try to move your mouse on the MESS window and to click on keys present on the artwork. The inputs shall be recognized by the system you are emulating and produce result as if you were pressing the key with your finger!

Media Emulation

In addition to the instructions below, see also the Software Lists section of this Manual for specific details about the “loading from software lists” option (which works like generic loading, except for requiring software to be named and stored exactly as expected by the .xml lists).

Shortcuts

<u>Tape Emulation</u>	<u>Quickloads</u>	<u>Printer Emulation</u>
<u>Cartridge Emulation</u>	<u>CD-ROM Emulation</u>	<u>Serial Ports</u>
<u>Floppy Emulation</u>	<u>Hard Disk Emulation</u>	<u>Choosing BIOS</u>

Tape Emulation

MESS supports tapes for a huge number of systems. Typical tape format for MESS is plain .wav: at loading time, any other tape format is internally converted to .wav and if you save any program to tape, you will obtain a .wav file.

To load a tape in MESS, you simply need to use the -cass option (for systems with many cassette drive, you use -cass1 or -cass2 respectively):

```
mess.exe <system> -cass foo.ext
```

Cassette loading is in general not automatic, except if it was on the original system. This means that you may need to enter CLOAD or LOAD or another command and then to press “Play” on the cassette drive. To press “Play” in MESS, you simply need to enter the Internal UI, choose the Tape Controls menu and press “Play”. The same procedure goes for any other tape operation (“Rec”, “Stop”, etc.).

To remove a tape in MESS or to replace it with a new one, while the emulation is running, you simply need to enter the Internal UI, choose the File Manager menu, select Cassette and click on the tape file you are currently using. This shall prompt you with a file browser which allows you either to change tape image or to select “empty slot” to remove completely the tape file from the cassette drive.

In the same file browser you can also choose to create a brand new cassette, choosing “Create”. In this case you have to enter a name and then a new wav file will be created, from the cassette drive. This is very useful when you want to save data or programs to a cassette. Again, the saving procedure is not automatic, except if it was on the original system, and you need to know the correct commands used by the emulated machine. At the end, you will find a .wav file in your MESS directory containing the data you wanted to save.

Notice that loading and saving procedures can be slow. However, you can speed up emulation with no bad consequences by pressing <key>Ins</key> in partial keyboard emulation mode.

Tips - MESS supports also a faster way to create a new tape file: from command line launch MESS with a non-existent .wav file

```
mess <system> -cass foo.wav
```

and the emulator will create an empty tape for you, without any need to enter the File Manager

Cartridge Emulation

MESS supports carts for a huge number of systems. To load a cart in MESS, you simply need to use the -cart option (for systems with many cartslots, you use -cart1 or -cart2 respectively):

```
mess.exe <system> -cart foo.ext
```

Cartridge loading is automatic once the emulation is started.

To remove a cart in MESS or to replace it with a new one, while the emulation is running, you simply need to enter the Internal UI, choose the File Manager menu, select Cartslot and click on the cart file you are currently using. This shall prompt you with a file browser which allows you either to change cart image or to select “empty slot” to remove completely the cart file from the cartslot.

If the original cart supported saving data to internal SRAM, MESS should save the same data to a nvram file foo.nv inside the folder nvram/<system>/. You can check [sysinfo.dat](#) to find if the system you want to emulate currently supports saving to SRAM.

Floppy Emulation

MESS supports floppy disk for a huge number of systems. To load a floppy disk in MESS, you simply need to use the -flop option (for systems with many floppy drives, you use -flop1, ..., -flop4 respectively):

```
mess.exe <system> -flop foo.ext
```

Floppy loading is in general not automatic, except if it was on the original system. This means that you may need to enter LOAD or another command to actually start floppy loading.

To remove a floppy disk in MESS or to replace it with a new one, while the emulation is running, you simply need to enter the Internal UI, choose the File Manager menu, select Floppy Disk and click on the disk file you are currently using. This shall prompt you with a file browser which allows you either to change disk image or to select “empty slot” to remove completely the disk file from the disk drive.

In progress - saving to a floppy - formatting/creating a floppy

Tips - MESS supports also a faster way to create a new disk file: from command line launch MESS with a non-existent file in a floppy drive

```
mess <system> -flop foo.ext
```

and the emulator will create an empty disk for you, without any need to enter the File Manager

Quickloads

MESS supports quickloads formats for a many systems. This kind of formats consist in a dump of the memory content of the system after a program or a game has been loaded. Re-loading the same data directly into the memory of the emulated system, you should be able to start immediately to use the program. To load a quickload in MESS, you simply need to use the -quick option:

```
mess.exe <system> -quick foo.ext
```

Once the Quickload file is in the memory, the program should started automatically.

To load a new quickload file in MESS, while the emulation is running, you simply need to enter the Internal UI, choose the File Manager menu, select Quickload and click on the quickload file you are currently using. This shall prompt you with a file browser which allows you to choose another quickload file.

No saving to quickload is implemented in MESS, nor it is planned addition of this. Saving memory contents can be already done in systems which supports State Saving, and other systems should add support for Save States rather than for Quicksave.

CD-ROM Emulation

MESS supports CD-ROM for a few systems. Typical tape format for MESS is CHD. You can read a guide about how to create a CHD from a CD [here](#)

[http://members.iinet.net.au/~lantra9jp1/gurudumps/tutorials/dumping_cds_&_hdds/index.html#cd], courtesy of The Guru. To load a CD-ROM in MESS, you simply need to use the -cdrom option

```
mess.exe <system> -cdrom foo.chd
```

CD-ROM loading is automatic once the emulation is started.

To remove a CD-ROM in MESS or to replace it with a new one, while the emulation is running, you simply need to enter the Internal UI, choose the File Manager menu, select Cdrom and click on the CHD file you are currently using. This shall prompt you with a file browser which allows you either to change CD-ROM image or to select “empty slot” to remove completely the CD-ROM file from the CD-ROM drive.

Hard Disk Emulation

In progress - loading - creating / formatting

Printer Emulation

In progress

Serial Ports

In progress

Choosing BIOS

Sometimes, MESS supports more BIOS dumps for a single system, e.g. to offer access to different OS revisions. To select among these, you can use the `-bios` option at launch. A list of available bios and related options for each system is available in [sysinfo.dat](#).

At the moment, there is no way to change BIOS while emulation is running.

Internal UI Options

During emulation in MESS, you can always press `<key>Tab</key>` to access the Internal UI and customize all the options you can need to fully enjoy your experience. This Internal UI works very similarly to MAME one (with a few additional items, as you will notice below). Hence, if you are used to MAME, you will immediately recognize what you need here.

Before passing to analyze each submenu of the Internal UI, please remember that if you enter the wrong submenu, you can always press `<key>Esc</key>` or scroll down to Return to Prior Menu to go back.

Input (general)

Exactly as in MAME, in this menu you can configure the UI controls and the common Inputs which will be shared by all the systems, e.g. the “Pause” key (by default mapped to `<key>P</key>`), the Snapshot key (by default mapped to `<key>F12</key>`) or the joypad/joystick buttons (e.g. Button 1 to 4 by default mapped to `<key>LCtrl</key>`, `<key>LAlt</key>`, `<key>Space</key>` & `<key>LShift</key>` as in MAME).

Notice that you must be careful when selecting the keys you want to use: they may be already in use for other functions.

The options changed in this menu will be saved in `cfg/default.cfg` (which can be opened with any text editor) and will affect every emulated system. If you end up setting something wrong and the emulation does not work anymore as you expect, **DON'T PANIC!** You simply need to delete the `default.cfg` file from the `cfg/` directory, in order to reset the settings to their default value. Every change can always be reverted in MESS!

Input (this System)

Exactly as in MAME, in this menu you can configure the Inputs which are system specific, e.g. the computer keyboards, or joypad/joystick buttons (e.g. Button 1 to 4 by default mapped to `<key>LCtrl</key>`, `<key>LAlt</key>`, `<key>Space</key>` & `<key>LShift</key>` as in MAME).

All you need to do now is to select the function you want to change, press enter and then press the key you want to use. It is possible to assign more than one key to a function by pressing enter again and pressing a different key. More fancy key combinations can be made as well:

- To map `<key>key1</key>` OR `<key>key2</key>` (i.e. more than one key will produce the same effect), set one of the keys as before, wait until MESS accepts it, then repeat for second key.
- To map `<key>key1</key>` AND `<key>key2</key>` (i.e. you need to press to keys at the same time to obtain an effect) press both (or more than two) keys after pressing enter.
- To map `<key>key1</key>` NOT `<key>key2</key>` (i.e. `<key>key1</key>` has an effect only if `<key>key2</key>` is not also pressed), select the function you want to change, press enter, press the first key and then quickly press the second key(s) twice.

The correspondent combination will appear on screen and you can check if the result is the one you wanted to obtain.

Notice that you must be careful when selecting the keys you want to use: they may be already in use for other functions. If you want to assign joystick movements to a function then once you have selected the

function you want to change move the joystick/pad or press the fire button you want to use (remember that, as explained above, you must have selected the -joystick option to activate the Joystick support if you want it to work in MESS).

The options changed in this menu will be saved in `cfg/<system>.cfg` (which can be opened with any text editor) and will affect only the chosen system and (possibly) its clones. If you end up setting something wrong and the emulation does not work anymore as you expect, **DON'T PANIC!** You simply need to delete the corresponding `.cfg` file from the `cfg/` directory, in order to reset the settings to their default value. Every change can always be reverted in MESS!

Dip Switches / Driver Configuration

Exactly as in MAME, in this menu you can setup some additional configurations which were available in the original systems. These may differ from system to system, or be completely absent.

The options changed in this menu will be saved in `cfg/<system>.cfg` (which can be opened with any text editor) and will affect only the chosen system and (possibly) its clones. If you end up setting something wrong and the emulation does not work anymore as you expect, **DON'T PANIC!** You simply need to delete the corresponding `.cfg` file from the `cfg/` directory, in order to reset the settings to their default value. Every change can always be reverted in MESS!

Categories

Here, for systems which support them, you can choose between different emulated controllers. Changing a setting in this menu, corresponds in some sense to unplug a controller from a controller port of the original system and to plug a new one into it. As an example, consider Nintendo Entertainment System (NES) which allows you to choose between a normal controller, a Zapper lightgun and an Arkanoid paddle in the Category Menu. Corresponding re-mappable inputs will appear in the Input (this System) menu after your choice here.

The options changed in this menu will be saved in `cfg/<system>.cfg` (which can be opened with any text editor) and will affect only the chosen system and (possibly) its clones. If you end up setting something wrong and the emulation does not work anymore as you expect, **DON'T PANIC!** You simply need to delete the corresponding `.cfg` file from the `cfg/` directory, in order to reset the settings to their default value. Every change can always be reverted in MESS!

Analog Controls

Here you can configure all the analog controls available for the system you're emulating (e.g. emulated mouses, paddles or lightguns). Many choices are given in this menu: you can make the controller more or less sensible in each direction, or invert the axis, to better suit your needs.

System Information

Here you can see some basic info about the system you're emulating (which system is, what CPU it used etc.). It simply shows you again the information screen that MESS had already showed at start.

Image Information

Here, you can see which image is currently loaded in each available device.

File Manager

Here, you can mount an image in one of the available media devices, or unmount the image currently loaded and mount a new one in the same media device or in a different one. This menu also allows you to create new images for devices which supports image creation and saving (e.g. cassette). You simply have to select

the device, and then the “Create” option. You will be asked to type in a filename and then the new image will be created (if there exists already a file with the same name, you will be asked if you want to overwrite the old one or to change filename).

Notice that, once you have selected the device you want to modify, the File Manager will allow you to browse every volume and directory of your computer and to select the file you want. Moreover, the File Manager allows you to use `<key>PgUp</key>`, `<key>PgDn</key>`, `<key>Home</key>` and `<key>End</key>` for a faster scroll and to input the first letters of the file you look for in order to reach immediately the image you want.

And if you want to load an image from the available software lists, you can do that as well by selecting the correspondent menu entry (more info on this topic can be found in the [Software List](#) Section of this Manual).

Keyboard Mode

Here, you can select in real time the keyboard emulation mode you prefer, between “Full Emulation Mode with Emulated Keyboard ” (emulated) or “Full Emulation Mode with Natural Keyboard” (natural). See [Emulated Keyboard](#) section of this Manual for more information about these two modes.

Slider Controls

In progress Here you find controls to stretch the screen and/or increase the contrast or brightness of the emulation. Also, if you activated cheats with the -cheat option, you can overclock the various CPUs of the system.

Video Options

In progress Here you can access Video options similar to those available for [MAME](#)

Cheat

If you have an xml cheat file in the directory, and you start [MESS](#) with -cheat (or setting cheat to “1” in [mess.ini](#), or selecting cheats in the [GUI](#)), here you will find the available cheats. Notice that cheat search is temporarily not available, as in [MAME](#).

Select New System

Here you can pass to emulate another system.

Return to System

Here, you exit from the menu to go back to the emulation!

Graphical Interfaces

There is no graphical user interface shipped with [MESS](#). You may be interested to use [MESS](#) in conjunction with some [frontend](#), if you are really scared by command line programs.

Video Options

Shortcuts

Overview	Video Parameters	Video Modes	Vertical Synchronization
--------------------------	----------------------------------	-----------------------------	--

Running MESS in a Window	Rotating Screen	Multiple Windows	Filters
Running MESS in Full Screen	Artworks	Stretching and Scaling	

Overview

Here we only analyze a few of the available video options for MESS. A more detailed description of the MESS video-related command line options and configuration options can be found at the [usage](#) and at the [mess.ini](#) pages of this wiki, respectively.

Running MESS in a Window

To run MESS inside a window, you can use the command line option `-window`. Alternatively, you can set

`window 1`

in [mess.ini](#) or untick the Fullscreen settings in GUI.

Also, notice that you can force MESS to start in a window which fills the screen by using the `-maximize` option.

Warning: for some reason `-maximize` does not work very well in (SDL)MESS under MacOSX. If you are on a Mac and you select to run in a window, you should also add `-nomaximize`, or set

`maximize 0`

in [mess.ini](#).

Running MESS in Full Screen

To run MESS in full screen, you can use the command line option `-nowindow`. Alternatively, you can set

`window 0`

in [mess.ini](#) or tick the Fullscreen setting in GUI.

Video Parameters

MESS let you customize brightness, contrast and gamma parameters. You can change these values in many different ways. They are present in the GUI, so you can tweak them before starting emulation. Alternatively, you can modify them in real time by entering the Internal UI (press <key>Tab</key>) and selecting the Slider Controls menu (see also the [Internal UI](#) section). Finally, at command line you can use the following options: `-brightness`, `-contrast` and `-gamma`.

-brightness

This affects how bright will be all colors, black and white included, during emulation. Notice that brightness parameter will affect also the artwork, if present. You can choose any decimal value between 0.1 and 2.0 , 1.0 being the default one. If you increase this parameters you will get a brighter screen.

-contrast

This affects the amount of difference between black and white, used by the screen. It only affects the emulated system, but not the Artwork or the Internal UI menus. You can choose any decimal value between 0.1 and 2.0 , 1.0 being the default one. If you increase this parameter, you will get a larger contrast between black and white.

-gamma

This affects the difference of color values between black and white. It only affects the emulated system, but not the Artwork or the Internal UI menus. You can choose any decimal value between 0.1 and 3.0 , 1.0 being the default one. If you increase this parameter, the middle value of a color will move down, giving more bright levels than dark ones, therefore increasing the overall brightness.

Rotating Screen

MESS allows the user to run emulation with a rotated and/or flipped screen. You can select these rotation options in many different ways. They are present in the GUI, so you can tweak them before starting emulation. Alternatively, you can modify them in real time by entering the Internal UI (press <key>Tab</key>) and selecting the Video Options menu (see also the [Internal UI](#) section). Finally, at command line you can use the following options: -ror, -rol, -flipx and -flipy.

-ror

This option is used to rotate the game screen to the right (i.e. clockwise)

-rol

This option is used to rotate the game screen to the left (i.e. counterclockwise)

-flipx

This option is used to mirror the game screen horizontally

-flipy

This option is used to mirror the game screen vertically

Artworks

In progress artwork options -aspect -resolution -view (also there are xxxxxN versions on Windows for separate screen settings!)

Video Modes

MESS allows you to choose also different video configurations, to improve compatibility with older video card. These settings are OS-dependent.

If you are using MESS for Windows, you can select

```
-video d3d  
-video ddraw  
-video gdi
```

D3D is the fastest setting, but it requires a Direct 3D 9+ compatible graphics card. If your card is not compatible with latest Direct 3D, you may want to try ddraw or gdi.

If you are using (SDL)MESS, you can select

```
-video opengl  
-video soft
```


OpenGL is the fastest setting, but it requires a OpenGL compatible graphics card and drivers. Alternatively, you can choose the slower but also more compatible soft rendering. You can also select to use your own alternative GL library by setting `-gl_lib /path/to/libGL.so` and select which SDL video drivers to use, by setting the `-videodriver` option to “x11”, “directfb”, etc. or to “auto” which uses the SDL default.

In every operating system, MESS also offers the option `-video none` which doesn't output anything on screen. This is very useful for benchmarking because it takes away any possible slowdown due to the graphics card.

Multiple Windows

MESS also allows to open the emulation in multiple screens, by setting the `-numscreens` option to an integer value greater than one. This will open multiple screens for the emulation, each one configurable to its own resolution etc. At the moment, (SDL)MESS only supports `-numscreen 1`.

Stretching and Scaling

You can force MESS to keep the aspect ratio of the emulated system despite any resizing of the window by using `-keepaspect` . In full screen mode, this will make black borders to appear to compensate your screen size vs. emulated screen size. If you turn this off (with `-nokeepaspect`), the emulated screen will be stretched to fill the whole window/full screen.

Also you can decide to scale the system graphics before displaying it. This can be done by using the option `-prescale <n>` with “n” an integer number (“1” is the default value)

(SDL)MESS also offers a few more scaling options, when in 'soft' video mode (see [Video Modes](#) section above). These can be selected by setting `-scalemode` to `async`, `yv12`, `yuy2`, `yv12x2`, `yuy2x2` or “none” (default value).

Vertical Synchronization

If you suffer from tearing or choppy scrolling, you can try to enable the `-waitvsync` option which will sync emulation to your monitor. This option works better in windowed mode.

Filters

MESS tries to reproduce the original aspect of the emulated systems. Hence, no fancy filters can be enabled. However, it offers a bilinear filter (active by default) though the `-filter` option. Using this in conjunction with an `-effect mask`, can produce a quite accurate representation of monitors and television screens. You can turn the bilinear filter to off with `-nofilter`.

Sound Options

Shortcuts

Overview	Audio Modes	Changing Volume
Muting MESS	Changing Samplerate	Changing Latency

Overview

Here we only analyze a few of the available sound options for MESS. A more detailed description of the MESS sound-related command line options and configuration options can be found at the [usage](#) and at

the [mess.ini](#) pages of this wiki, respectively.

Muting MESS

You can easily deactivate sound in MESS by using `-nosound` option. Notice that this option does not stop emulation of the sound hardware, it only mutes MESS output to your sound card.

Audio Modes

In (SDL)MESS you can decide which SDL audio driver to use through the `-audiodriver` option. You can select “alsa”, “arts”, etc. or leave “auto” for SDL default driver.

Changing Samplerate

You can set the sound frequency used by MESS setting `-samplerate` to your preferred value. The higher the number the better the sound quality. However, sometimes this comes at expense of the performances on older systems. Hence, you can set it to something lower than the default 48000, e.g. to 11025, 22050 or 44100, and see if it better suits your needs.

Changing Volume

You can change the volume, measured in db, of the sound emulation by using the `-volume` option. This accepts an integer value (even a negative one!) which represent the number of times the volume is doubled/halved. E.g.

```
mess.exe -volume 3
```

will produce a volume which is doubled three times compared to the default.

```
mess.exe -volume -5
```

will give a volume which is halved five times compared to the default.

You can modify the volume also in real time, by pressing `<key>Tab</key>` and entering the `Slider Controls` menu.

Changing Latency

If you get crackling sound in the emulated system, you can try to modify audio latency which sets how much buffering MESS is allowed to do when emulating sound. The default value is `-audio_latency 2` but you can try to increase it to reduce glitches or decrease it to improve responsiveness. However, before fiddling with this option, check first

1. that you are running emulation at 100% (i.e. your computer is fast enough for the system you are emulating)
2. that the emulated system is expected to have NO sound issues

Quite often imperfect sound is due to lack of CPU power or issues in the emulation.

Controller Options

Shortcuts

<u>Supported Controllers</u>	<u>Automatically Enabling Specific Controllers</u>	<u>Keymaps</u>
<u>Using Joysticks and Mice</u>	<u>Joymaps</u>	
<u>Multiple Input Devices</u>	<u>Analog Deadzone</u>	

Supported Controllers

This entry refers to controllers on the user system. For emulated controllers, i.e. the controllers that were used in the emulated machines, please refer to [this](#) entry and [this](#) entry. You may also want to read [this page](#) for a general discussion about input devices.

The simple fact is, if Windows supports your joystick or gamepad and it has been set up properly in the Windows Control Panel, there shouldn't be an issue with MESS supporting it. If you haven't configured your controller, then open the Start → Settings → Control Panel window. In Windows 2000 and XP you will have a system Controllers icon. You'll have to refer to the Windows help file to configure your controller, but make sure that the main controller is “Joystick 1”.

A similar situation holds true for (SDL)MESS: if the SDL libraries recognize your controllers, the emulator will recognize them as well!

Using Joysticks and Mice

Once you have a joystick, a mouse or a lightgun recognized by your operating system, you can use it in MESS by simply activating it at start. This can be done in various ways. If you are using messui.exe in Windows, you can activate Joystick, Mouse and Lightgun from the correspondent Input settings.

Alternatively, you can use at command line the options -joystick, -mouse and -lightgun, or modify the corresponding configuration options in [mess.ini](#).

Namely,

- You can activate Joystick support adding the -joystick option at command line, or setting to “1” the joystick line in [mess.ini](#)
- You can activate Mouse support adding the -mouse option at command line, or setting to “1” the mouse line in [mess.ini](#)
- You can activate Lightgun support adding the -lightgun option at command line, or setting to “1” the lightgun line in [mess.ini](#)

Once your controller has been activated, its inputs will be recognized in both “Input (general)” menu and “Input (this System)” menu of the Internal UI, and you will be able to remap emulated axis and keys to your controller.

Multiple Input Devices

In MESS you can separate inputs from multiple keyboards and mice, by using the options -multikeyboard and -multimouse *In progress*

Automatically Enabling Specific Controllers

In MESS there are also options which allow to automatically enable keyboard, mouse or joystick when a particular class of devices is used by emulated system. You can obtain this effect by setting the command line options

```
-paddle_device
-adstick_device
-pedal_device
-dial_device
```

```
-trackball_device
-lightgun_device
-positional_device
-mouse_device
```

to one of the allowed values “none”, “keyboard”, “mouse”, “lightgun”, “joystick”. E.g. to map dial inputs to a joystick you can use

```
-dial_device joystick
```

Notice that your settings would be ignored if, in this case, you have no joysticks connected to your computer.

Joymaps

The `-joystick_map` option allows to control how joystick values map to digital joystick controls. MESS accepts all joystick input from the system as analog data. For true analog joysticks, this needs to be mapped down to the usual 4-way or 8-way digital joystick values. To do this, MESS divides the analog range into a 9×9 grid. It then takes the joystick axis position (for X and Y axes only), maps it to this grid, and then looks up a translation from a joystick map. This parameter allows you to specify the map. The default is 'auto', which means that a standard 8-way, 4-way, or 4-way diagonal map is selected automatically based on the input port configuration of the current game. Maps are defined as a string of numbers and characters. Since the grid is 9×9, there are a total of 81 characters necessary to define a complete map. Below is an example map for an 8-way joystick:

```
777888999 Note that the numeric digits correspond to the keys
777888999 on a numeric keypad. So '7' maps to up+left, '4' maps
777888999 to left, '5' maps to neutral, etc. In addition to the
444555666 numeric values, you can specify the character 's',
444555666 which means "sticky". In this case, the value of the
444555666 map is the same as it was the last time a non-sticky
111222333 value was read.
111222333
111222333
```

To specify the map for this option, you can specify a string of rows separated by a “.” (which indicates the end of a row), like so

```
777888999.777888999.777888999.444555666.444555666.444555666.111222333.111222333.111222333
```

However, this can be reduced using several shorthands supported by the `<map>` parameter. If information about a row is missing, then it is assumed that any missing data in columns 5-9 are left/right symmetric with data in columns 0-4; and any missing data in columns 0-4 is assumed to be copies of the previous data. The same logic applies to missing rows, except that up/down symmetry is assumed. By using these shorthands, the 81 character map can be simply specified by this 11 character string: `7778...4445`. Looking at the first row, `7778` is only 4 characters long. The 5th entry can't use symmetry, so it is assumed to be equal to the previous character '8'. The 6th character is left/right symmetric with the 4th character, giving an '8'. The 7th character is left/right symmetric with the 3rd character, giving a '9' (which is '7' with left/right flipped). Eventually this gives the full `777888999` string of the row. The second and third rows are missing, so they are assumed to be identical to the first row. The fourth row decodes similarly to the first row, producing `444555666`. The fifth row is missing so it is assumed to be the same as the fourth. The remaining three rows are also missing, so they are assumed to be the up/down mirrors of the first three rows, giving three final rows of `111222333`.

Analog Deadzone

MESS also allows to specify a joystick deadzone for analog inputs which do not center properly, by using the `-joystick_deadzone` option. Setting e.g.

```
-joystick_deadzone 0.4
```

will produce a dead area (where no joystick movements are detected) equal to 40% of the joystick movement. The value specified can vary between 0.0 and 1.0

Keymaps

SDL only, for non-US keyboards *In progress*

Cheat Options

Shortcuts

Activate Cheats	Cheat Format
---------------------------------	------------------------------

Activate Cheats

You can activate cheats in MESS by selecting them in GUI or by using the `-cheat` option at command line. Unfortunately, cheats are currently not working in MESS after MAME 0.127 cheat rewrite.

More precisely, you can add cheat in xml format as `system.xml` but you cannot currently add per game cheats as in previous versions. They should be re-added soon.

Cheat Format

When re-added, cheats will be in the new xml format MAME is using after 0.127. More info will come when support is added back.

Save Options

Shortcuts

NVRAM of the Emulated Hardware	MESS Save States
--	----------------------------------

NVRAM of the Emulated Hardware

For system which had an internal backup system, or for carts which were able to save progresses in SRAM, MESS offers the same feature by using `.nv` files inside the `nvrasm/` directory. These files contain the same data the original system (or cart) would have saved.

SRAM support is not always perfectly supported. You can check [sysinfo.dat](#) for more details about support in the various systems.

MESS Save States

`-state`
`-autosave`

In progress

Snapshot Options

Shortcuts

Save a Snapshot	Snapshot Naming Options
---------------------------------	---

Save a Snapshot

To save a snapshot (a.k.a. screenshot) of an emulated system in MESS, you simply have to press `<key>F12</key>` during emulation. The image will be saved in .png format in the `snap/` directory. The user can customize the resolution of the snapshot through the option `-snapsize` (either at command line or in `mess.ini` or in the GUI). Also, for games with an artwork, the user can decide if the artwork must be included or not in the png. This can be chosen by using the option `-snapview`. E.g. for pc1403 setting the default option

```
-snapview internal
```

will produce a snapshot also including the artwork. Setting

```
-snapview "Standard (4:3)"
```

will produce a snapshot without the artwork (only the digit display on a black background). The `-snapview` setting needed to exclude the artwork can be found in the Video Options menu of the Internal UI, since its name corresponds to the View Settings which turn off the artwork.

Remember that, by default, the snapshot is saved with a progressive number into a subfolder named `system/` (which MESS creates if there is none). In other words, if you take a snap of MESS running pce emulation, you will find a .png file as follows “`snap/pce/0000.png`”. Following snapshots will be named in progressive order “`0001.png`”, “`0002.png`” etc. However, the naming rules can be partially modified by the user, as described in [next entry](#).

TO DO: Better explain the views and stress that they are the same ones used in the `-view` command and that they can be found by loading a game, pressing tab, selecting video options and checking out the views in the list. The view used can also be different from the one you've got displayed.

Snapshot Naming Options

To customize the way snapshots are saved in the `snap/` folder, MESS offers the option `-snapname`. As of MESS 0.142, you can choose various combinations of the following templates

- `/` = path separator
- `%g` = system name
- `%i` = index
- `%d_xxxx` = device xxxx (xxxx being the brief name of a media device)

As an example,

```
mess.exe adam -snapname %g
```

will save your first Coleco Adam snap as “`adam.png`” inside the `snap/` folder, and then it will overwrite it whenever you press again `<key>F12</key>`. On the other hand,

```
mess.exe adam -snapname %g%i
```

will save snapshots in the `snap/` directory with progressive names “`adam0000.png`”, “`adam0001.png`”, etc. The default setting is

```
mess.exe adam -snapname %g/%i
```

which produces a subfolder named adam/ which will contain all snapshots, and the snaps will be named as “0000.png”, “0001.png”, etc.

Moreover, starting from MESS 0.142, it is possible to use the %d_xxxx template to name snaps after the name of the software file mounted in the media device xxxx. E.g. you can use “-snapname %d_cart” to use the name of the game mounted in the cartslot, or “-snapname %d_flop2” to use the name of the program mounted in the second floppy drive, or “-snapname %d_hard1” to use the name of the hard disk mounted in the first HD drive, etc. Remember that you can find out which media device are supported by each system by using the -listmedia option.

If you are loading by fullpath, e.g.

```
mess.exe nes -cart "Super Mario.nes" -snapname %d_cart
```

pressing <key>F12</key> will create a “Super Mario.png” file in the snap/ directory. While if you are loading through software lists, then its shortname will be used, e.g.

```
mess.exe nes -cart smb -snapname %d_cart
```

will result in a “smb.png” file in the snap/ directory.

Of course, any combination of the %d_xxxx template with the old /, %g and %i templates is supported as well.

Notice that if you specify an invalid -snapname template, or if you specify an invalid device (like %d_cass for a system without tape support), or if you have no software mounted in the chosen media device, then MESS will revert to the default template %g/%i.

Recording Options

MESS offers various different recording options. You can record

- video directly in .avi format
- video in .mng format (basically, a sequence of .png files)
- audio in .wav format
- inputs, to allow playback of great play

Each different recording alternative is controlled by a specific option. Recall that you can combine these options. E.g. you can first record a great play as an .inp file, without any slowdown due to the ongoing recording, and then use the playback function to record an .avi of the play, which can be shared afterwards.

Shortcuts

Recording AVI Movies	Recording MNG Movies	Recording WAV	Recording Inputs
--------------------------------------	--------------------------------------	-------------------------------	----------------------------------

Recording AVI Movies

You can record .avi movies of MESS emulation by using the -aviwrite option at command line. The resulting movie will be output in the snapshot folder (snap/ by default) as <system>.avi. Notice that the output is an uncompressed .avi file and can be very large (hundreds of MBs for less than a minute of video, in some cases)!

Also .avi recording is influenced by -snapsize and -snapview options

Recording MNG Movies

You can record .mng movies of MESS emulation by using the -mngwrite option at command line. The resulting movie will be output in the snapshot folder (snap/ by default) as <system>.mng. Notice that .mng files do not have audio. Hence, you may want to use this option in combination with the audio recording -wavwrite

Also, .mng recording is influenced by -snapsize and -snapview options, hence take a look to the [Snapshot Options](#) section of this Manual.

Recording WAV

You can record a .wav file of MESS emulation by using the -wavwrite option at command line. The resulting audio recording will be output in the same directory as MESS as <system>.wav.

Recording Inputs

You can record a .inp file of MESS emulation by using the -record option at command line, followed by a <name>.inp parameter. The resulting .inp recording will be output in inp/ directory as <name>.inp. You can replay the .inp file by using the -playback option, followed by the <name>.inp parameter of the .inp file you want to playback.

TO DO: verify which snap options influence .avi and .mng recording

System Specific Options

Please refer to sysinfo.dat or to the [MESS System Information File](#) section of the MESS Wiki to find instructions on how to use various media devices in MESS emulated systems.

MESS Tools

Shortcuts

chdman	casstool	imgtool	wimgtool
------------------------	--------------------------	-------------------------	--------------------------

chdman

In progress

casstool

In progress

imgtool

In progress

wimgtool

In progress

MESS Error Messages

Shortcuts

Missing Files	Software Not Found	Device Not Found	Wrong System Name
-------------------------------	------------------------------------	----------------------------------	-----------------------------------

Missing Files

If you receive a

```
<file> NOT FOUND  
ERROR: required files are missing, the system cannot be run.
```

message, you need to find the files which are missing.

Software Not Found

If you receive a

```
Device Cartslot load () failed: File not found
```

message, or a

```
Device Floppy Disk load () failed: File not found
```

message, or a similar one, you need to check if you gave the right path for the software.

Device Not Found

If you receive a

```
Error: unknown option: -carts
```

message, you need to check which device options you have used vs. which device options are supported by the system (using `-listmedia`)

Wrong System Name

If you receive a

```
"<system>" approximately matches the following  
supported systems (best match first):
```

message, you need to check what is the correct name of the system you want to run.