pfe

pfeMAME

Manual

Author
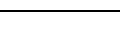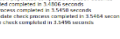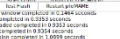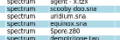
Gareth Finch
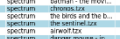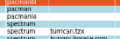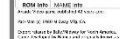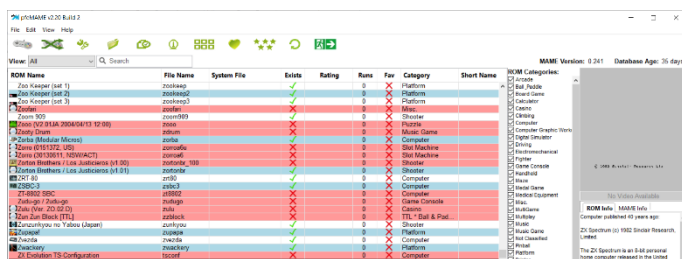
Table of Contents

1 ABOUT

pfeMAME is a cross-platform MAME front-end written in (wx)Python. It has been tested to run on Linux (Ubuntu), WSL2-Ubuntu, Windows 8 / 10 / 11, and MacOS, but should be able to run on any platform supporting Python and wxWidgets as long as it is 64 bit.

As of version 2.25, pfeMAME has been updated to work with Python 3.10 and wxPython 4.2.2 as a minimum. The versions must also be 64 bit as 32 bit is no longer supported.

There are notable differences in the Linux / Windows / MacOS versions due to the way that the different platforms handle graphics. That said, the differences are minimal. There are still some ongoing challenges in the different widget support under wxPython on the various platforms and this has led to some possible odd behavior.

2 SUPPORTED VERSION OF MAME

pfeMAME currently requires a minimum MAME version of 0.206 to run reliably. Earlier versions may cause issues due to additional features added to MAME and the support for these features in pfeMAME.

3 MERGED ROMSETS

Merged romsets are currently not supported by pfeMAME. So, if for example you have merged your clone romset into the main romset, pfeMAME will not be able to find it, so it wont be populated in the ROMs list.

As per a number of other people on the Internet, the simple solution is to create a 0-byte file named as the clone set. So, for example, if you have all of your puckman and clones in puckman.zip and you want to run pacman, create a empty text file and rename it to pacman.zip. Then refresh the database and the ROM will be listed. When you run it, MAME will check for missing files automatically inside the parent file which in this case is puckman.zip.

For pfeMAME to automatically work this out itself, I would have to firstly add the parent, then look up the XML info for the parent, then look for all children related to that parent, work out what files the children need, then check inside the parent ZIP file if those children files are present, and if they are then add that child / clone to the ROMs list. That is a huge overhead in processing, and I'm really not interested in looking at it for now. I may look into this in the future, but it would slow things down a lot. HDD space is no longer a premium guys.

4 PREREQUISITES

 a. If running the source Python version (i.e. not a packaged EXE version);

  • Python 3.10 (64 bit) or newer

- wxPython phoenix 4.2.0 or newer (not classic)

- pypubsub (load with pip install pypubsub)

- python-docx (load with pip install python-docx)

- python-vlc (loaded with pip install python-vlc)(Although this is not necessarily needed – if you don't have it the application will fall back to wx.MediaCtrl)

- If using VLC, you need the 64 bit version

- Optionally, if using VLC you can use ffprobe to retrieve the video file meta data for the fullscreen video player. You will need to manually download ffprobe and place it in the same folder as the pfeMAME executable.

b. A version of the MAME command line executable (version 0.206 and above)

c. The MAME INI file must be configured correctly at least for the rom file path. If you can run the MAME exe and play a ROM then the rest will work ok.

d. An up-to-date copy of the following (Just get the versions matching your version of MAME). You will need to put them in the same folder as the MAME executable in Windows. For Linux, put them in the root location where all of your other MAME support folders are (like roms, samples, etc);

- history.dat or history.xml (The XML version is preferred)

- catver.ini

e. Some MAME compatible roms

5 DEPENDENCIES

5.1 WINDOWS

The following DLL files are required for the Windows distribution, and it is assumed that they are present on your system. The common missing file is MSVCP90.dll which comes with .NET. All missing DLL files can be found on the internet if required. Unfortunately, I can't distribute them with pfeMAME due to licensing restrictions.

OLEAUT32.dll
USER32.dll
COMCTL32.dll
SHELL32.dll
ole32.dll
WINMM.dll
WSOCK32.dll
COMDLG32.dll
ADVAPI32.dll
WS2_32.dll
WINSPOOL.DRV
GDI32.dll
MSVCP90.dll
KERNEL32.dll
RPCRT4.dll

5.2 LINUX

The distribution version of pfeMAME requires that the same or newer version of GLIBC be on the Linux system that you run it on. If an older one exists, then you will likely end up with a runtime error. To fix this you must update your Linux system to the latest version or build pfeMAME yourself.

6 INSTALLING

6.1 INSTALLING UNDER WINDOWS

There is nothing special required for installing under Windows. Just copy the files to a location that you like and run the pfeMAME.exe file. There is a possibility that you will get an error saying that it can't find a dll of some kind. Please see the dependencies section of this manual for help.

6.2 INSTALLING UNDER MAC OS

There is nothing special required for installing under Mac OS. Extract the pfeMAME.app from the downloaded ZIP/7z file and double click to execute it. The operating system may ask you to grant permissions depending upon the folder you have copied it into.

NOTE: For best 'stable' results, I suggest you don't lauch by double clicking on the pfeMAME.app file for now. Right click and select 'Show Contents. Then navigate as shown below and double click on the pfemame executable file. This is recommended due to ongoing issues with the way the .app environment is being setup that I have not worked out yet.
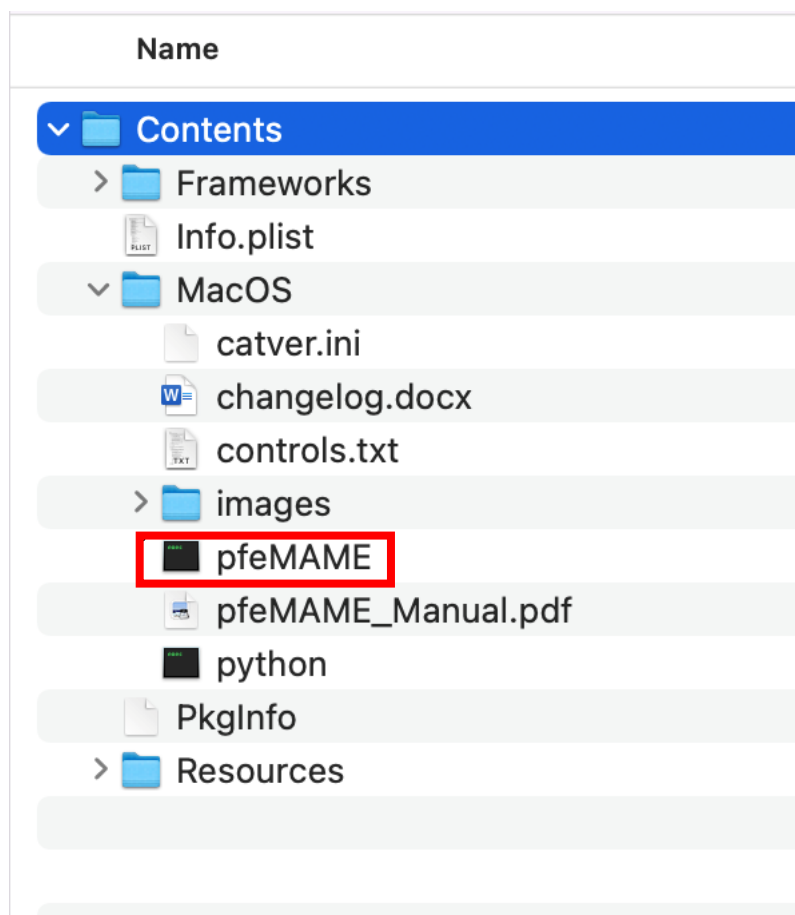
6.3 INSTALLING UNDER LINUX

Installing under Linux can sometimes require setting file permissions before running. This goes for distribution and source code versions. If you're getting any issues with getting pfeMAME to run, set all files

in the pfeMAME folder to have read/write access for all users. Different distros have different ways of doing this from the GUI – in Ubuntu you can run Nautilus as SUDO from a terminal and then right click the pfeMAME folder and set all permissions to read/write.

You may also need to set the pfemame executable file to be executable by running chmod u+x pfemame from the command line.

Run the application using ./pfemame from the command line.

If you get an SDL not found error, install the library using sudo apt-get install sdl1.2-dev.

It can be difficult to find the MAME executable file in Linux. Under Ubuntu it is typically located at /usr/games. The auto-find feature under file paths will try and work this out for you.

The MAME INI file is typically located in a hidden folder in your home directory called .mame (See the dot in front of the name).

Sometimes when copying the distribution version of pfeMAME to your Linux distro, you have to right click the pfeMAME executable and select it to 'allow executing file as program'.

I have also seen an issue when the pfeMAME application runs ok, but when you try to run a game rom the following error is displayed in the messages log window; **/lib/libstdc++.so.6: version `GLIBCXX_3.4.15' not found** . If you get this error then there is a library mismatch between the distribution version and your distro. In this situation, the only way to rectify is to download the pfeMAME source code and build it yourself. See the chapter later in this manual on how to build from source.

7   FIRST-TIME RUN

The first time you run pfeMAME you need to select the language, and then set up the default file paths and any preferences. NOTE: The language can be later changed from the pfeMAME preferences window.

7.1   EDIT > FILE PATHS

This sets up the paths / folders for various support files and core application type files. As a minimum, the first 8 should always be set up, although the application will work without Icons and Samples.

To manually select a folder / path, double click on the row to open a file / folder selection window.

There is an Auto Find feature that will ask you to select the root folder for MAME, after which it will look for all logical folder matches and populate those that it finds. It will also look for the support files. In Linux it also checks the /usr folder for the MAME executable.

The MAME INI file location is critical. As of pfeMAME 2.6 the MAME file paths and preferences are now being stored in the mame.ini file. MAME then retrieves these variables when it runs. MAME will always look for its mame.ini file in a very specific location on Windows and Linux. On Windows its always to be in the same root folder as the MAME executable. In Linux, it 'should' be in the users HOME directory in a folder called '.mame'. This is a hidden folder in Linux and is created when you install MAME. You don't have to worry about creating the mame.ini file (Which MAME can do if you ask it to) – the pfeMAME application will create it if it can't find it. When you first run pfeMAME, it will look for the mame.ini file in the locations where it should be, and will show that location in the Edit File Paths window. If that location in blank then you need to do some investigating as things are not as they should be.

Please note that the Auto Find feature will not locate files for other emulation engines (model2/3 and Visual Pinball) as they can be pretty much anywhere.

As of pfeMAME 2.9, the file paths window supports multiple folders for some folder types. The types are pretty much defined by MAME itself. Folders like ROMS and SAMPLES can have more than one path. If a folder does support multiple paths, then you will see an 'Add' option next to it. Click on 'Add' and an additional row will be created. Then double click on that row to select the folder path. Additionally, you can delete the additional folder paths by clicking on the 'Delete' option.

Edit File Paths

Name	Path	Add	Delete	Clear
MAME INI File:	C:\Mame\mame.ini			Clear
MAME Executable:	C:\Mame\mame64.exe			Clear
history.dat file:	C:\Mame\history.dat			Clear
Snap folder:	C:\Mame\snap			Clear
Icon folder:	C:\Mame\icons			Clear
Roms folder:	C:\Mame\roms	Add		Clear
Software folder:	C:\Mame\roms			Clear
Video folder:	C:\Mame\video			Clear
Samples folder:	C:\Mame\samples	Add		Clear
Softwarelist HASH folder:	C:\Mame\hash	Add		Clear
Plugins folder:	C:\Mame\plugins	Add		Clear
catver.ini file:	C:\Mame\catver.ini			Clear
Artwork folder:	C:\Mame\artwork	Add		Clear
Controller folder:	C:\Mame\ctrlr	Add		Clear
INI folder:	C:\Mame\ini	Add		Clear
Config folder:	C:\Mame\cfg			Clear
Mapping folder:	C:\Mame\mapping			Clear
NVRAM folder:	C:\Mame\nvram			Clear
Difference folder:	C:\Mame\diff			Clear
Cheat File folder:	C:\Mame\cheat	Add		Clear
Inputs folder:	C:\Mame\inp			Clear
States folder:	C:\Mame\sta			Clear
VP Tables folder:	Double click to set			Clear
VP EXE File:	Double click to set			Clear
m2 Emulator EXE File:	Double click to set			Clear

OK   CANCEL   CLEAR ALL   AUTO FIND   MAKE INI

7.2 PREFERENCES

These are the system wide default preferences for the application. The MAME related preferences are in the other tabs. The preferences window is in a separate tab on the main screen.
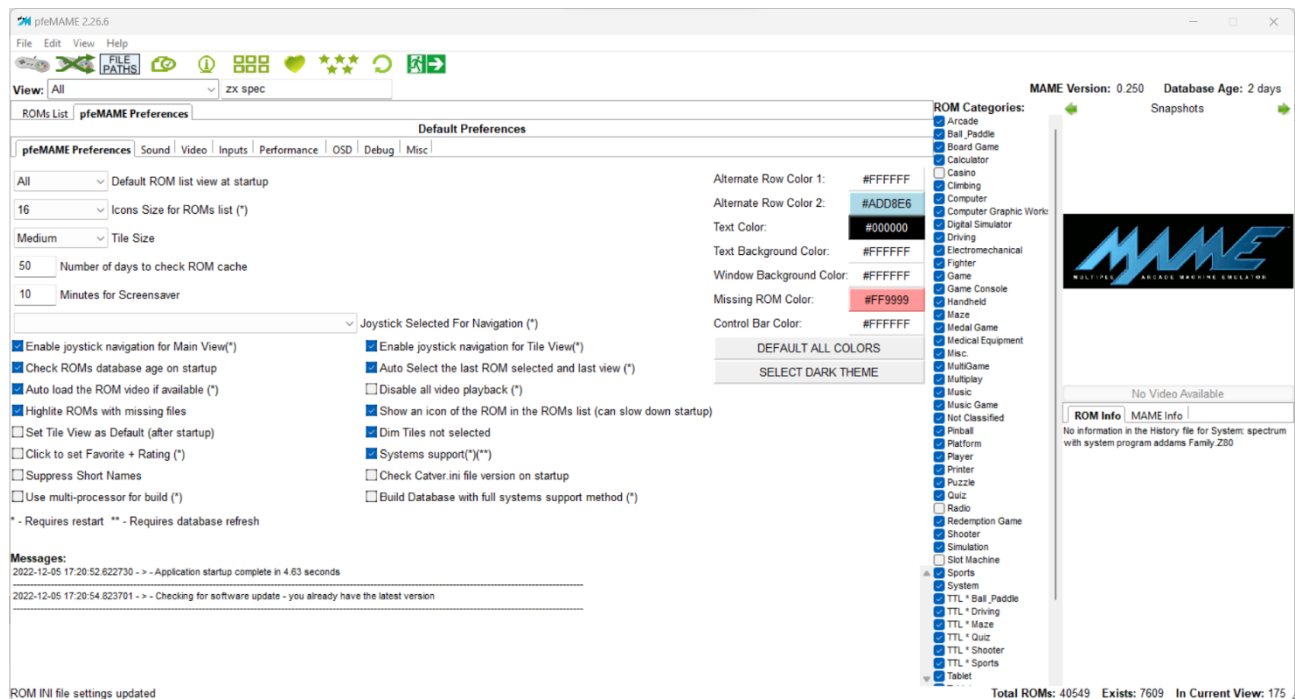
7.3 ROM CATEGORY FILTER

As of pfeMAME version 2.3, the ROM category selections are on the main window (see below). These allow you to select what ROM categories are shown in the main window. To select all, double click in a free space in the list, and likewise do the same to de-select all.

Game Categories:
- ☑ Ball _Paddle
- ☑ Board Game
- ☑ Calculator
- ☑ Casino
- ☑ Climbing
- ☐ Coin Pusher
- ☑ Computer
- ☑ Driving
- ☐ Electromechanical
- ☑ Fighter
- ☑ Game Console
- ☑ Handheld
- ☑ Maze
- ☑ Medal Game
- ☑ Medical Equipment
- ☑ Misc.
- ☑ MultiGame
- ☑ Multiplay
- ☑ Music
- ☑ Not Classified
- ☑ Pinball
- ☑ Platform
- ☑ Printer
- ☑ Puzzle
- ☑ Quiz
- ☑ Rhythm
- ☑ Shooter
- ☐ Slot Machine
- ☑ Sports
- ☑ System
- ☑ Tabletop
- ☑ Telephone
- ☑ Utilities
- ☑ Whac-A-Mole
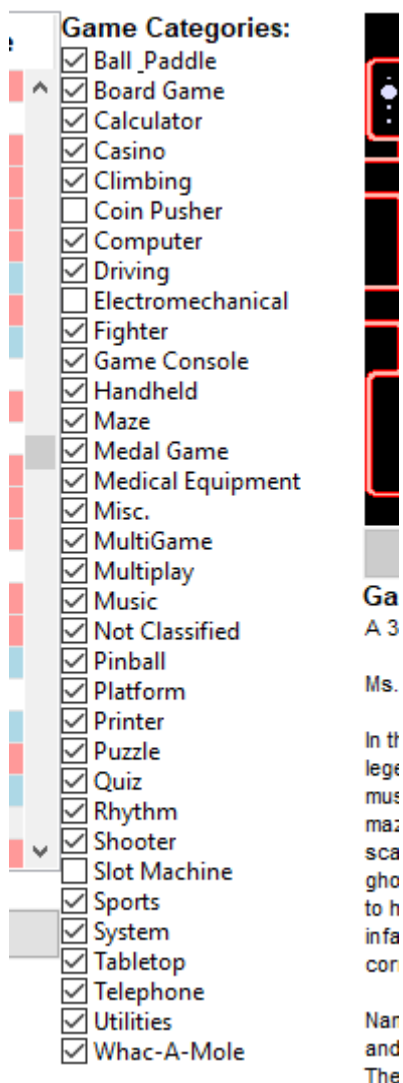
Once the preferences and file paths are set up, restart pfeMAME. If needed, manually refresh the ROMs database (click on the refresh toolbar icon or select VIEW > Refresh ROMs Database). Refreshing the database will force a full read and set up of the database which is then saved in a cache file to speed up the application startup next time. The application will check the age of the cache based upon the setting in 'Preferences' and will recommend that you manually refresh the database to keep this up-to-date (definitely do it if you add / change roms, etc). You should be presented with something that looks like this;

8 BUILDING THE ROMS DATABASE

The first time you run pfeMAME, after setting up the necessary preferences etc, if a valid ROMs database doesn't already exist, the application will automatically create one. Once a database is created, every time you restart the application, it will load the database from a cache file.

In preferences you can set the number of days to check for updating the cache (Number of days to check cache file).

You can manually force a database refresh at any time by pressing F5 or selecting the refresh icon on the toolbar.

When the database is refreshed, there are some additional options in the preferences as follows.

Systems Support – This allows the application to look for systems programs in the MAME ROMs folder. Systems are things like home computers and consoles – basically hardware that loads other programs. In the MAME ROMs folder, you create folders called the same as the system program, however some of the names need to be quite specific and can be found in the XML file (HASH file) for that particular MAME rom. For example, ZX Spectrum home computer programs need to be stored in folders called spectrum_cass (and a few others). You need to work this out yourself as MAME doesn't provide any hints.

As an example;
> For the ZX spectrum program 'The Birds and The Bees'; The program file name is 'the birds and the bees.tzx', while the short name in the XML file is 'birdsbee'. Your ROMs folder structure needs to look like this;

- ROMS
  - spectrum_cass
    - birdsbee
      - the birds and the bees.tzx

> If you were to structure as per the above, and from the command line run;

> Mame.exe spectrum birdsbee

> then MAME will execute the program using the settings stored in the XML file. When you refresh the pfeMAME database it will assume the program is stored as per above, but if you have not stored it correctly it will still try to find it for you but executing it with XML HASH file support likely won't work.

All the necessary info is in the XML file so happy hunting.

Use multi-processor for build – This tells the application to create a pool of worker objects when building the ROMs database, and to feed each worker to a separate CPU or core. It can be faster on slower systems, but really comes into its own when used with the following setting (Lets just say that it's pretty much compulsory in this instance). NOTE: THIS IS CURRENTLY EXPERIMENTAL AND I HOPE TO IMPROVE THE PERFORMANCE IN TIME WHEN A NEW RELEASE OF PYTHON SUPPORTS SHARED MEMORY WITH CHILD PROCESSES.

Build database with full systems support method – this is a bit hard to explain, but when MAME looks for folders to find systems programs, as mentioned above it looks for the name in the system XML file. The tag in the file is 'softwarelist name='. In my first implementation of systems support, I only allowed for folders called the same as their systems name, or with a suffix after an underscore. This works for a lot of systems, but not all. Some systems also expect to see programs in a totally different named folder. The only way to support that was to look for the folder name in the XML file for every system. This is the thorough method but is very slow – on my computer it can take 700 seconds. When you enable the multi-processor build above, this can drop substantially (mine dropped to 180 seconds). This depends on many factors, including the number of processors / cores and how fast your file system is (SSD will be much faster than a mechanical drive).

9 RUNNING A ROM

There are multiple ways to run a ROM once everything is set up. NOTE: You must have actually selected a ROM in the window and that ROM must actually exist.

- From the FILE menu, select 'Play ROM' or 'Play Random ROM'

- Pressing the 'ENTER' key

- Right click the ROM to bring up the context menu, and select

    o 'Play ROM'

    o 'Play ROM without softlist HASH support'

    o 'Play ROM with different engine'

10 CONTROLLER INPUT MAPPING

MAME supports controller input mapping – which is great for setting the various buttons on your gamepads, etc. to work nicely with a game you are running. The problem is that it is quite generic in nature and doesn't allow you to have different configurations for different games.

You can mess around with creating custom controller files for various bios or system types but I find that's very messy and, at least for me, quite confusing.

As of version 2.15.4, pfeMAME supports per-ROM input mapping via what I think is a nice and easy method.

Firstly, you must create a 'Mapping' folder within your MAME folder. This isn't a standard MAME folder and is instead used by pfeMAME. I could have put it within the pfeMAME distribution but I dint like that idea.

In pfeMAME, use File Paths to select the 'Mapping' folder you created.

Now run a ROM file and define the input mapping using the MAME built-in Tab menu and Input Mapping. Once you are done, exit MAME. Then in pfeMAME, right click the ROM you have just run and select 'Copy controller mapping to this ROM'. If you have other ROMs that you want to use the same mapping, then select each one in turn and do the same.
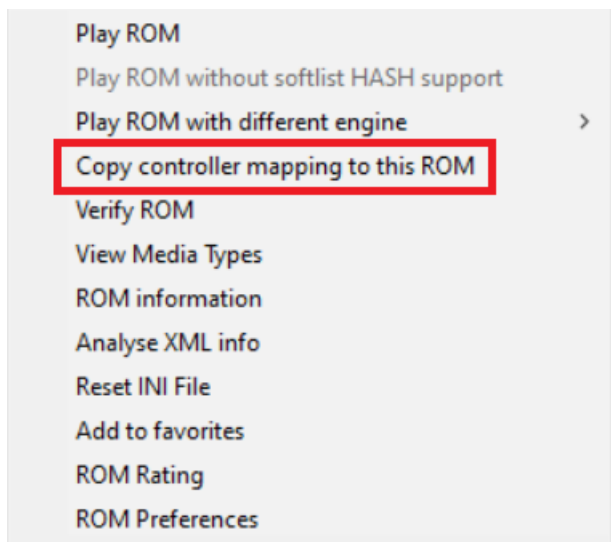
What the above process is doing is copying the default.cfg file that MAME created when you first defined the Input mapping within MAME. The file is renamed to match the ROM name and stored in the Mapping folder. Then when you next run that ROM, the file is copied back across and overwrites the default.cfg file.

Now with the process above you can have different games with different input mapping just by selecting another ROM, defining its inputs differently as you require, then ruing the right click menu to copy its mapping for later.

Easy – I hope!

11 KEYBOARD SHORTCUTS

The following is a list of keyboard shortcuts that can be accessed via the main window.

ESC	This will close the active window (Except for Tile View where you must use the '2' key and the main window where you must use CTRL-Q)
CTRL-P	Brings up the default pfeMAME preferences
CTRL-SHIFT-P	Brings up the ROM specific preferences
CTRL-Q	Quits pfeMAME from the main window
CTRL-F	Find a ROM
SHIFT-V	Launch the full screen video player
F5	Refresh ROM database

12 TILE VIEW

This is an alternate view for the application primarily designed for stand-alone MAME cabinets. It provides an image 'tile' of the ROMs. You can navigate with the arrow keys or a joystick / gamepad. It defaults to showing the ROMs you have flagged in your favorites list, although you can toggle to the full (existing) list – but obviously a very large list will be hard to navigate.

NOTE: If no ROMs are visible then check that you have assigned any to your favorites list.

The pfeMAME application can be set up to automatically launch into Tile View after startup by checking the appropriate box in the preferences section – this again is designed with MAME cabinets in mind.
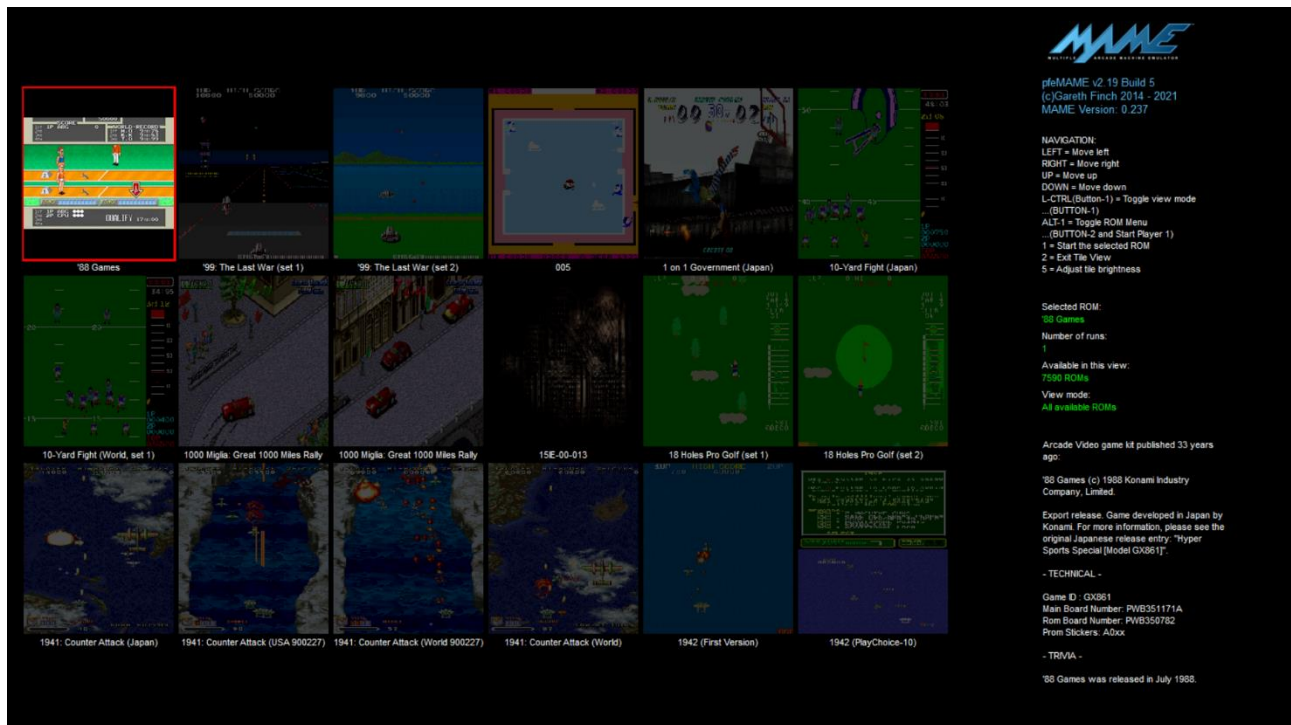
A note about the mouse while in Tile View; As this view is designed for MAME cabinets with joysticks, use of the mouse is assumed to be a 'no go'. Unfortunately, wxPython doesn't have a global way of disabling the mouse pointer. What I have done is to turn the mouse pointer into a blank image when it hovers over any control. This sort of works, although not all controls seem to support this feature so if you move the mouse around you will see it at times. In addition, DO NOT USE THE MOUSE TO CLICK ON ANYTHING. If you do it may cause the 'tiles' to lose focus or you will change the focus to a spot the program doesn't understand. This can make life difficult when you try to close the window.

13 STAR RATING

Right click a ROM and select ROM Rating'. This allows you to rate your ROMs (0 ~ 5 with 0 being default). You can also double click in the rating column to bring up the star rating prompt (if this has been selected in preferences).

14 FAVORITES

Right click a ROM and select 'Add to Favorites'. You can then filter by favorites using the 'View' drop down box. You can also double click in the favorites column to add / remove a favorite flag (if this has been selected in preferences).

15 NUMBER OF RUNS

This displays the number of times a ROM has been run. It only considers a ROM 'run' if you actually load all the way into the ROM.

16 SEARCHING

To search for ROM name, rom file name, or system file name, just type your search into the search field. You can use CTRL-F to set focus in the search box. It uses a 'type ahead' search, so it dynamically resizes the list as you type. As the search is performed on the full ROMs list it also sets the 'View' drop down filter to 'ALL' as soon as you start typing.

You can enter a series of search words that don't have to be in a specific order in to get the right match – useful if you don't exactly know the right format of a long name e.g.

looking for rom name: blah blah (bootleg of whatever)

search for: blah bootleg

17 SORT ORDER

By default, the application will show the ROMs list sorted by the ROM Name. You can change this to sort by any column by clicking on that columns header.

18 BACKUP AND RESTORE

Under the File menu you can select to backup or restore various application files. This is useful if moving from one platform to another and you want to for example back up your favorites.dat file. The application will search for all of the DAT files and the pfemame.ini file and back up what it can find. You have to select an existing folder to back up to. When restoring it will ask you to specify a folder and it will restore all of the relevant DAT files and the pfemame.ini file that it finds.

19 ROM VIDEOS

If rom videos are found in the 'videos' folder, a message under the ROM image will say 'Video Available – Click to play'. Clicking this message will load and play the video over the top of the ROM image. If in the pfeMAME preferences you have selected to auto-play the video, then it will load and automatically play 5 seconds after selecting the ROM. The video file format must be a valid one for the operating system and the OS must have a valid video backend available (e.g., gstreamer for Linux, WMP10 for VISTA/7/8/10).

The video file will be played over the top of the ROM image.

If the video doesn't play, e.g. just a black window, check the video format. Try opening the video using the operating system's default video player (Media Player under MS Windows). If it doesn't work there, then it's not the right format.

By default, pfeMAME will try to open the video using VLC media player as the backend. This requires you to have VLC 64 bit version installed on your system. Currently this is not supported under Linux or MacOS due to some rather annoying issues on those OS's.

20 VIDEO PLAYBACK

There are two modes of video playback supported by pfeMAME. The primary playback is through VLC (If it is installed on the system and it's the 64 bit version) and the backup / secondary method is using the wx MediaCtrl.

NOTE: At present due to issues with wxPython support for VLC under Linux and Mac OS, on those platforms it will default to the wx MediaCtrl. I am not sure exactly what the issue is at this stage. I don't think it's wxPython itself but more likely the python-vlc library trying to use X windows and the issues with that implementation.

NOTE2: Further to the above, on Linux if the Wayland screen manager is detected rather than X then video playback will be disabled across the system. If you have the 'option' of using X on your OS version, then you can force the pfeMAME application to be run using the X screen manager by using the command below (from the command line). I believe this can also be hardcoded into a Linux equivalent of a shortcut but that is beyond the scope of this manual. The success and stability of this can be variable as it is running xwayland which just isn't very good.

GDK_BACKEND=x11 python3 pfemame.py -forcevid

NOTE3: If you get a 'playbin' error in Linux, this means that you have a problem playing back videos on your platform.

VLC playback supports a great variety of video file formats while wx MediaCtrl file support is limited to the backed for that system and what CODECs it supports. I'm finding that MP4 for example on Windows won't play back using wx MediaCtrl. Basically, if you can't play the file in the native OS video playback software (e.g. Media Player on MS Windows) then it won't play back in pfeMAME.

If you are finding issues launching pfeMAME due to video playback issues or the playback is corrupted, you can start the application with the -novid option and then in Preferences you can disable video playback.

21 FULL SCREEN VIDEOS

The application has a setting to launch a full screen video player (If a valid video playback backend exists) after a set period of time. This acts like a screen saver. The videos played are random videos stored in the 'Videos' folder.

The playback time setting is found on the main Preferences window called 'Minutes For Screensaver' and the default is 10 minutes.

During playback, pressing the SPACE key will skip to the next video. Any other key will exit.

22 Joystick Navigation

If a joystick or gamepad is connected to the system, it can be used to navigate on the main screen and within Tile View. Button 1 will run the selected ROM.

By default, on first installation no Joystick is selected. This can be changed from the main preferences window.

The bumper buttons on a gamepad will navigate up / down a page in the Main View.

In Preferences you can selectively enable joystick navigation in the Main and Tile views. For Tile View if you are using a cabinet with USB controller mapped to the cursor keys, you may want to disable joystick navigation support if you have any issues.

pfeMAME includes support for software list systems / home computers / consoles etc. with the inclusion of MESS into MAME. It is a selectable option in the preferences settings to support systems (by default when first installed this is disabled – once enabled you must restart pfeMAME and refresh the ROMs database).

To use the system support you MUST have a catver.ini file that includes systems in it. This is the only way that pfeMAME can tell that a file is a system.

When building the ROMs database, if pfeMAME finds a file that is listed as a system, it then looks in the ROMs folder for a subfolder with that systems name. If one exists, it then indexes all files in the subfolder and adds them to the database. To find programs for that system, just search for the system name, or the program name. For every system program that it finds, it then does a search in the hash folder XML file for that system to see if the program is listed. If it is listed, then it gets the program short name. The short name is needed to support soft list cheats for systems – this is an aspect of system support that is hardcoded in MAME and that's just the way it works.

NOTE: In the Tile View, systems will only work properly as 'favorites' - if you try running a system program under the 'All' view then it will run the top-level system (e.g. ZX Spectrum ROM) but not the system program – this is because under the 'All' view it will see the base system name first and just run that.
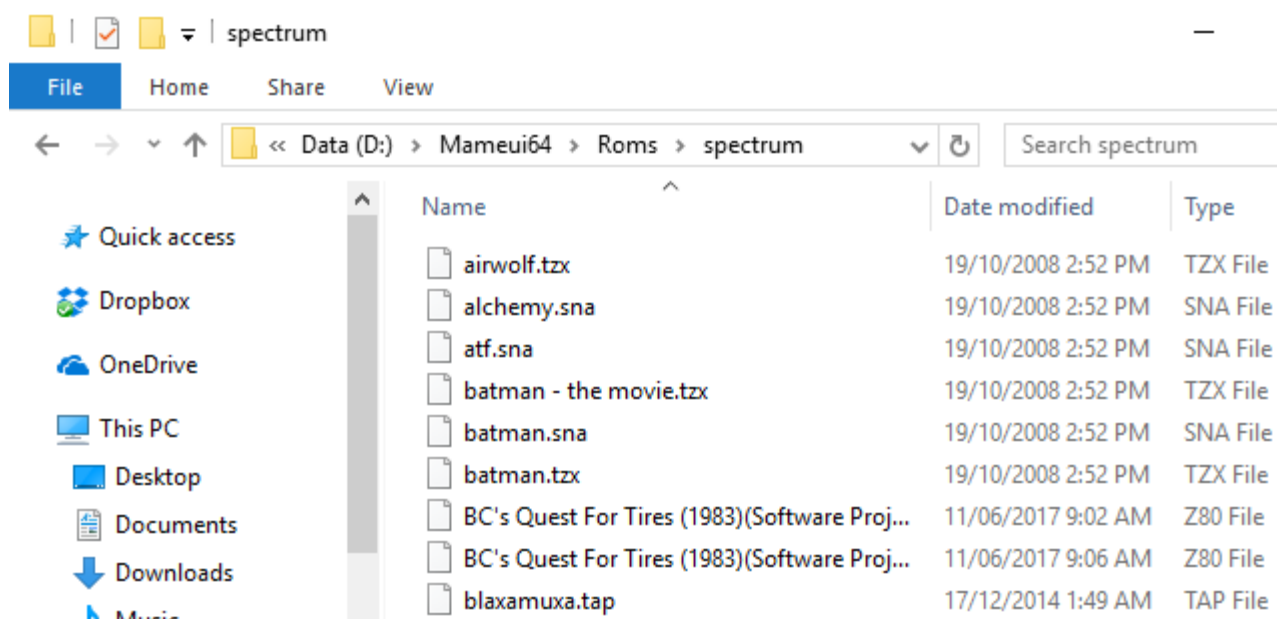
As systems are based upon MESS, they try to look up an XML 'hash' file stored in the HASH folder (distributed with MAME). These XML files list all details about the various programs / ROMs for the particular system. As systems (especially home computers) can have literally thousands of different programs and ROMs, many have not been put into these HASH files. pfeMAME looks for the hash file for the particular system program. If it can find it then it stores the short name for that program. When running the program, it uses the short name – this allows cheats to be supported for systems.

NOTE: Some system programs just currently won't work with the short name and come up with an error that it can't find the file. The reason this usually happens is because the file you have is not matching what

is specified in the XML file. It may be the file size or CRC is incorrect. You can check this by right clicking the file and selecting Analyse XML Info. This will tell you if anything is mis-matched with the file. If you want to bypass using the short name, then just right click the program and select 'Play ROM without softlist HASH support'. You can also select 'Suppress Short Names' from the pfeMAME Preferences.

As of pfeMAME version 2.3, system slots can also be configured on a per-rom basis. See the system slots section for more info.

24 ALTERNATE EMULATION ENGINES

As of pfeMAME 2.27 the alternate emulation engine support feature has been redesigned allowing expansion to any other emulation engine that supports command line running. I won't go into the details of how to install and use other emulation engines, but at a basic level just set them up and make sure that on their own they work and can run ROMs before you try to integrate them into pfeMAME.

Alternate emulation engines are supported in two different modes:

- Supplemental
- Additional

24.1 SUPPLEMENTAL MODE

This mode is for alternate emulation engines that support the same ROMs as MAME. It may be that their emulation is more playable due to the nature of MAME being more focused on the accuracy of the hardware emulation first, and speed second.

In Supplemental mode, the engine has specific ROMs associated so that pfeMAME knows that when that ROM is selected to run, it uses the alternate emulation engine rather than MAME. This can be overridden on a case-by-case basis by selecting 'Play ROM with MAME rather than alternate engine' from the right-click menu as shown below.
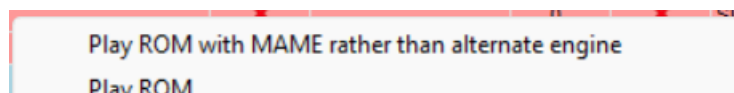
24.2 ADDITIONAL MODE

This mode is for emulation engines that emulate systems that MAME does not. A good example is Visual Pinball. Visual Pinball doesn't just emulate the pinball ROMs, it also provides a graphical play area that MAME cannot.

In Supplemental mode, the ROMs (or in the case of Visual Pinball the Tables) are listed separately in the main ROMs list. See the example below for Visual pinball.

| ROM Name	File Name	System File	Exists	
Virtua Tennis 2 / Power Smash 2 (Rev A...	vtennis2		✗	
Virtual Boy	vboy		✗	
Virtual Mahjong 2 - My Fair Lady (J 9806...	myfairld		✗	
Virtual On Oratorio Tangram M.S.B.S. ver...	vonot		✗	
Virtual Ping Pong (Conny / SDW Games)	conyping		✗	
Virtual Ping Pong (Protocol)	protpp		✗	
Virtual Pool	virtpool		✗	
Virtual TV Ping Pong	vtvppong		✗	
Virtual TV Soccer	vtvsocr		✗	
Virus (Dr. Mario prototype, PlayChoice-10)	pc_virus		✗	
Visor Edge	visor		✓	
Visual Memory Unit	svmu		✓	
Visual Pinball	Visual Pinball	Addams Family 7-1	✓	
Visual Pinball	Visual Pinball	Addams Family	✓	
Visual Pinball	Visual Pinball	Aladdin's Castle (Bally 1976)	✓	
Visual Pinball	Visual Pinball	Aladdin's_Castle	✓	
Visual Pinball	Visual Pinball	Amazing_Spiderman_The_1.26bob	✓	
Visual Pinball	Visual Pinball	Andromeda	✓	
Visual Pinball	Visual Pinball	Asteroid Annie (Gottlieb 1980) 2.0	✓	
Visual Pinball	Visual Pinball	Bad Cats 3.0.1	✓	
Visual Pinball	Visual Pinball	Batman - DE - LH 4-3 - V1.01	✓	

24.3 SETTING UP ADDITIONAL EMULATION ENGINES

From the main menu, select EDIT -> Setup Alternate Emulators. This will present the screen below.
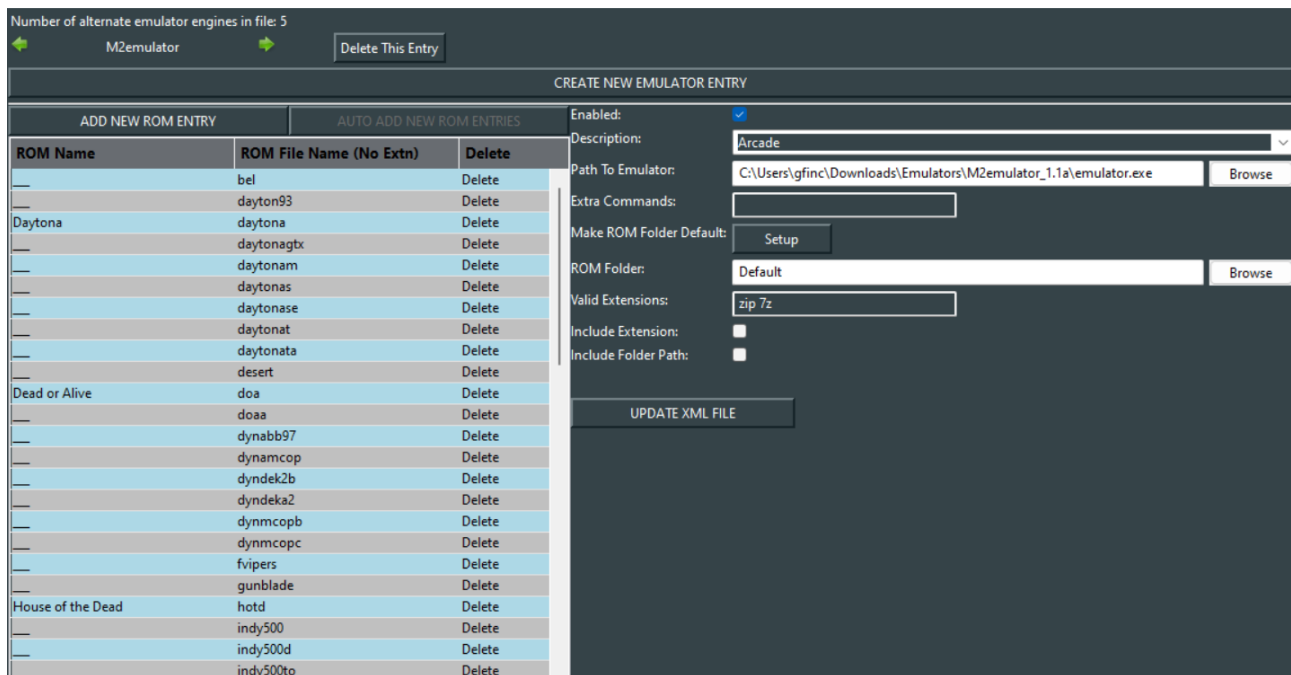
By default there is nothing entered. To create a new entry carry out the following;

- Click on 'Create New Emulator Entry'. When prompted, give your emulation engine a name.
- Next you have to set up the options for this engine.
- Tick the 'Enabled box'. Later you can un-tick this to disable this emulator globally.
- From the drop down list, select the 'Description' that best matches.
- Browse to and select the executable file for the emulator.
- Enter any required command line entries – for example, '-play'. These are usually used to tell the emulator to run the selected ROM or to run full screen. If you have multiple commands then separate them with a space.
- Set the ROM folder. If it is the same ROM folder as MAME (i.e. this emulator uses the same ROMs as MAME) then click the 'Setup' button to just make the ROM Folder 'Default'. If not, then select the folder where the ROMs for this alternate emulator are stored. In the case of Visual Pinball you actually want to select the 'Tables' folder.
- Enter valid extensions allowed for the ROMs / Tables (e.g. zip, 7z, vpt, vpx…). Separate additional extensions with a space. Do NOT include the decimal point '.' before the extension.
- Select 'Include Extension' if this emulator requires you to specify the extension when running the ROM (e.g. a command line may look like 'jaguar.exe – play daytona.zip').
- Select the 'Include Folder Path' if the emulator requires that in the command line. This is usually for emulators that don't have their own INI file specifying where the ROM is located (or you have not set it up) but just require it to be specified in the command line.
- Click on 'Update XML File' to store your settings.
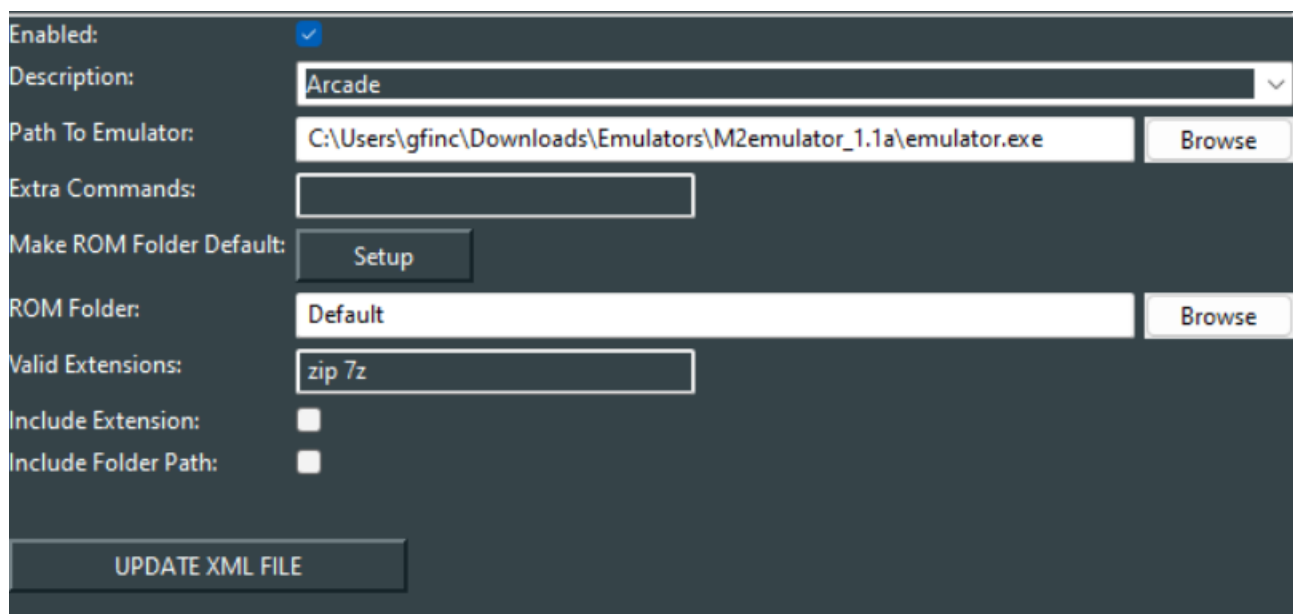
- Next you may need to specify the actual ROMs associated with this emulator. If the ROMs / Tables are in their own folder then just click on 'AUTO ADD NEW ROM ENTRIES'. Note, that is only available if you have specified a ROMs folder. This will add all entries from that folder into the ROMs list. Alternatively you can just add a single entry by clicking on 'ADD NEW ROM ENTRY'. Once all entries are complete you need to update the XML file again.

You can test the setup by clicking on the 'Test ROM' button.

Once you have set everything up, return to the main screen and refresh the database. If you have followed the steps correctly your new emulator engine should be available.

Below I have shown some example setups for various alternate emulators.

M2Emulator (For Sega Model 1 & 2)

Supermodel3 (Sega Model 3)

Enabled:	☑
Description:	Arcade ▼
Path To Emulator:	C:\Users\gfinc\Downloads\Emulators\Supermodel_0.3a-git-6dcf144_Win64\su Browse
Extra Commands:	
Make ROM Folder Default:	Setup
ROM Folder:	Default Browse
Valid Extensions:	zip 7z
Include Extension:	☑
Include Folder Path:	☑

UPDATE XML FILE

Jaguar (Atari Jaguar)

Enabled:	☑
Description:	Game Console ▼
Path To Emulator:	C:\Users\gfinc\Downloads\Emulators\BigPEmu_v104\BigPEmu.exe Browse
Extra Commands:	
Make ROM Folder Default:	Setup
ROM Folder:	c:\mame\roms\jaguar Browse
Valid Extensions:	j64
Include Extension:	☑
Include Folder Path:	☑

UPDATE XML FILE

Visual Pinball X (Pinball Tables)

Enabled:	☑
Description:	Pinball ⌄
Path To Emulator:	C:\Visual Pinball\VPinballX.exe [Browse]
Extra Commands:	-play
Make ROM Folder Default:	[Setup]
ROM Folder:	C:\Visual Pinball\Tables [Browse]
Valid Extensions:	vpx
Include Extension:	☑
Include Folder Path:	☑

[UPDATE XML FILE]

25 LOG FILE

Located in the pfeMAME folder, the log file (logfile.log) contains useful startup and debug information. When the logfile gets too large the application will automatically create additional log files with a file suffix of .1, .2, .3, etc. You can view this logfile within pfeMAME by selecting VIEW > View pfeMAME Log File.

26 INI FILES

The way pfeMAME uses the MAME and ROM INI files is detailed below.

As of pfeMAME 2.6, the application utilises the MAME INI file for storing all MAME specific preferences, file locations, etc. The application will look for the mame.ini file where it should be when you set the file paths.

ROM specific preferences are stored in the INI folders path (In windows this is usually in the same location as the MAME executable). You can create these INI files manually, or change the ones created by pfeMAME. pfeMAME will always maintain existing information in these files and only overwrites preferences values that pfeMAME supports.

Being able to add your own options in the ROM specific INI files is very useful as you can support options that are not included in pfeMAME.

27 PREFERENCES

As per the above, pfeMAME supports default preferences, and ROM specific preferences. Default preferences can be found in a separate tab on the main screen. ROM specific preferences can be found by right clicking a ROM name and selecting ROM Preferences – a new tab will appear and be automatically selected. The two different views are shown below, including a view with slot configurations for systems.
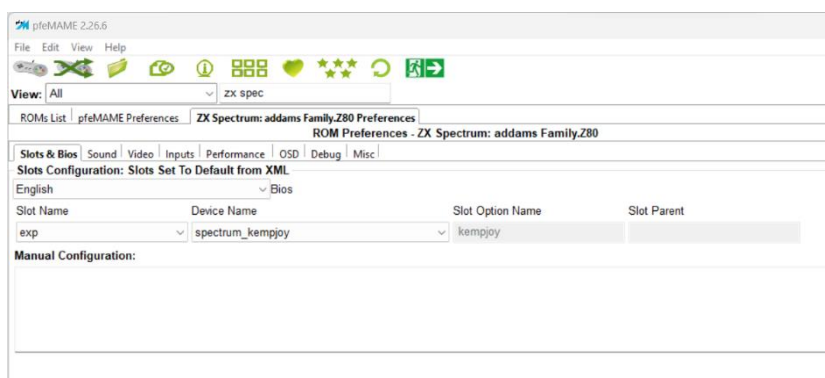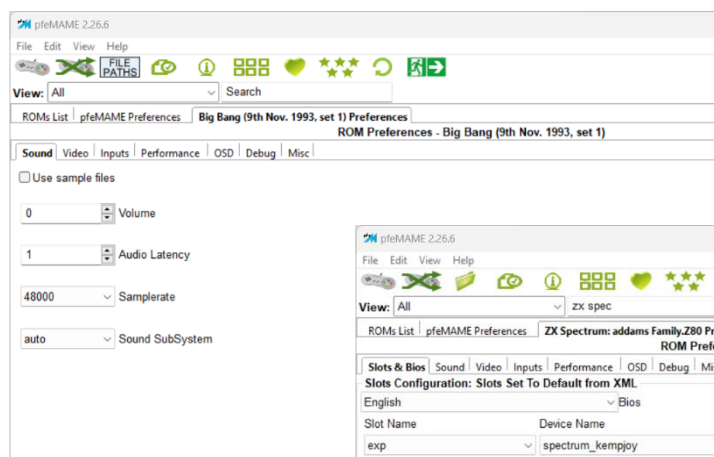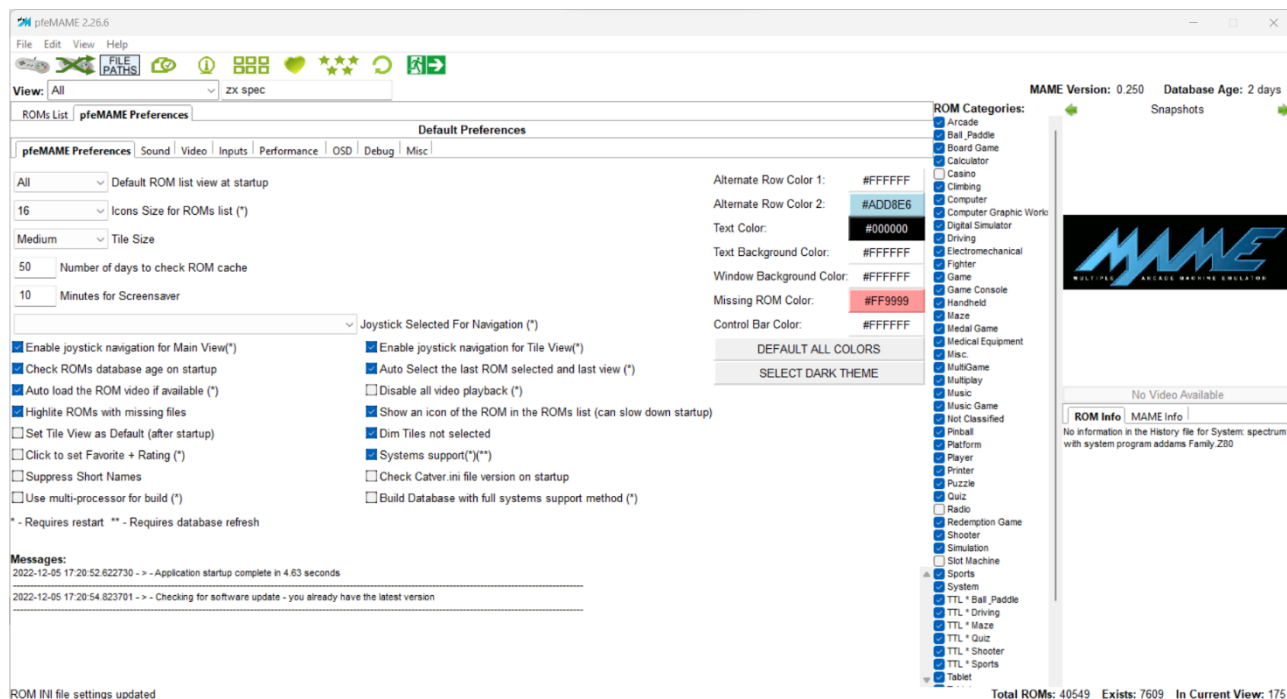
27.1 ROM PREFERENCES

ROM preferences allow you to set preferences that are specific to a rom file. This includes systems (e.g. ZX Spectrum, C64, etc), and even the programs that you run on those systems. A ROM specific INI file will be created and stored in the INI folder (This must be defined in the file paths window). When MAME runs the rom, it loads this INI file and any preferences found will take priority over the default preferences. This means that you can personalise any ROM preferences and have it remember those preferences for next time.

27.2 SLOT CONFIGURATION & BIOS SELECTION

Slots

This allows you to store system program specific slots (e.g. computer expansion cards). These can be saved per-system program, so for example different ZX Spectrum programs can have different slot configurations saved to take advantage of add-on-modules that provided speech, sound, joystick, etc.

There is an option to reset the slots to empty – this will erase any slot information and set the number of slots available based upon the XML information for the system type. There is also the option to reset the slots to default – this is similar to the reset slots to empty except that it erases the slots information and then loads any default settings that are noted in the XML file – this feature is not currently working perfectly (as of version 2.8) due to some consistency issues in how slots data are defined and used.

The Slots Configuration heading will make it clear if the slots have been loaded from the ROM INI file or set based upon default data in the XML file, or if they are just empty (No default info in the XMl file).

It is important to note that the XML file I refer to is created by MAME by running the -listxml option. This is all managed automatically by pfeMAME.

Bios

On the Slots & Bios tab there is a Bios drop down box that allows you to select the Bios (or in the case of old 8-Bit computers the actual ROM) that will be run when you run the ROM. As an example, it allows you to select Kickstart 3.1 ROM for the Amiga 500.

If nothing is found in the ROM specific INI file then the drop-down will default to the first Bios in the list.

27.3 MANUAL CONFIGURATION

Any additional configuration options specific to a ROM or system can be stored here. This is useful for options that are not currently supported in pfeMAME, as well as cascading slots hardware.

pfeMAME supports multiple languages, but its not perfect. I'd say more of a work-in-progress.

When the application is run for the first time you are asked to select your language.

Once selected, if English is not your default language then you will need to restart the application for the selected language to take effect. You can change this selection later from the pfeMAME preferences window.

Language support is managed through pythons built-in features. The language files are stored in the 'locales' folder as per below.
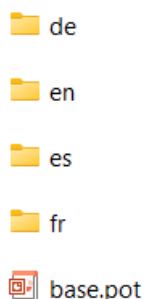
I have used Google to perform automatic translations as I don't speak other languages and don't know anyone that can do this for me. If you're not happy with the tranlations (and I am sure they are not great as Google doesn't really understand context) then you can update them yourself. You need to edit the 'base.po' file that is located within the language folder. For French it would be here.

locales

-> fr

  -> LC_MESSAGES

    -> base.mo

    -> pase.po

The 'base.po' file is just a text file. You will find the English versions of messages listed with the translated versions underneath. Once updated, you need to run the 'create_mo_file.bat' file. This will convert the PO file into the binary MO file that python needs. NOTE: the batch file is only available from the

distribution source. Can I please ask that if you do update the translations that you send the updated translation file to pfemame@gmail.com so that I can update for the next release.

29 COMMAND LINE OPTIONS

When running pfeMAME, either from the development source or from the distribution build EXE files, you can include some command line arguments as per below.

pfemame -d

This puts the application into developer mode. This displays additional staged startup times of various parts of the application. This is used for startup improvements and finding bottlenecks.

pfemame -r *romname*

This causes the application to run the defined romname after startup is complete.

pfemame -novid

This suppresses all video playback features. This is useful for platforms that don't support video playback or if you are having issues with it and cannot get to the preferences menu to switch the playback feature off in time to prevent issues from arising.

pfemame -novlc

Disables use of the VLC backend for video playback and forces use of the internal mediactrl (which is horrible and lacks support for a lot of video formats).

pfemame -v

Displays the pfeMAME version.

pfeMAME -l *language*

Sets the application to the desired language (English, Spanish, German, French)

pfemame -h

Displays the available command line options.

30 DAT FILES

pfeMAME uses a number of DAT files to store information about ROMs. These consist of the following;

- runs.dat
- favorites.dat
- ratings.dat
- enginetouse.dat

You must not manually edit these files unless you are sure you know what you are doing. The most important thing is that the very last line should be blank, i.e. there must be a carriage return or linefeed after the very last line.

31 SOURCE FILES

Source files for pfeMAME are available from the sourceforge page. NOTE: As of version 1.009, pfeMAME only supports python 3.4 and above and the wxPython phoenix widgets.

32 LINUX FILE LOCATIONS

It can be difficult to find the MAME executable file in Linux. Under Ubuntu it is typically located at /usr/games. In the case of SNAP installs of MAME it could be in a few different places but almost always exists in the system $PATH variable so you can just execute it at the command line using 'mame'.

The MAME ini file is typically located in a hidden folder in your home directory called .mame (See the dot in front of the name).

33 Reporting problems, bugs, issues, improvements

All suggestions for changes / improvements are welcome (No guarantee of implementing them though).

If you find a bug or problem, please do report it along with the following;

- Screen shot if possible
- Detailed description of what you saw / what happened and any events leading up to it
- Copy of the following;
  - logfile.log
  - mylist_temp.cfg
  - pfemame.ini
  - mame.ini
  - categories.dat
  - favorites.dat
  - ratings.dat
  - runs.dat

34 The Windows platform and graphics / icons

The python and wxPython applications under windows seem to be less tolerant to graphical file issues which can lead to application crashes. I have tried to include as much error checking as possible to trap or circumvent these problems but at times they may still occur. The common problem is if your icon (.ICO) or ROM image (.PNG) is malformed or the wrong format. Under Linux this is handled fine, and you should not notice any major problems. Under Windows I'm just not sure. I have found many application crashes and have implemented a number of workarounds / error traps but it may still occur.

35 Building pfeMAME from source

Building from source has been tested using python3.6, and python3.7, and python 3.8.

35.1 Installing cx_Freeze for Windows and Linux

Before you can build a distribution from the pfeMAME source, you need to install cx_Freeze. There are a few methods to do this as follows;

METHOD 1: Using PIP

pip install -U cx_Freeze

METHOD 2: Download the wheel for your version of python

This example is assuming you have python 3.7 and a 64-bit operating system. Download the cx_Freeze wheel listed below – NOTE: Always go for the latest version of cx_Freeze;

- cx_Freeze-5.1.1-cp37-cp37m-win_amd64.whl

METHOD 3: Download from GIT

This method seems to be the most reliable. Firstly, you have to go to GIT and downlad it. Then open a GIT bash prompt and execute the following;

pip install --upgrade git+https://github.com/anthony-tuininga/cx_Freeze.git@master

NOTE:

If you are installing cx_Freeze under Linux and you encounter the following error;

/usr/bin/ld: cannot find -lz

collect2: error: ld returned 1 exit status

error: command 'x86_64-linux-gnu-gcc' failed with exit status 1

You are missing a file – install it using;

sudo apt install zlib1g-dev

35.2 Installing py2app for Mac OS

On Mac OS I am using py2app to create and application bundle. This can be installed with the following command;

 pip3 install py2app

35.3 Building for Mac OS

For Mac OS builds I am using py2app. From the command prompt make sure you are in the source code folder and run the command below;

 python3 setup.py py2app

This will create a build & dist folder. Within the dist folder you will find the pfeMAME.app bundle. You can run it by double-clicking on it.

35.4 Building for Linux

For Linux builds, as per the Windows instruction below, you can use cx_Freeze. To install cx_Freeze, use the following;

    python3 -m pip install cx_Freeze --upgrade

Open a terminal window, change to the source folder you wish to build and execute the following (it is suggested you copy your source folder to a temporary folder so that you don't add unwanted build files to your main source folder);

        python3 setup.py build

The deployment package will be transferred to a subfolder named 'build'. To run this under Linux, open a command prompt to the location of the files, and use the './pfemame' command to run the application.

35.5 Building for Windows

To build pfeMAME from source under windows, you must have the following installed;

- Python3
- wxPython phoenix
- cx_Freeze
- pywin32 extensions

Open up the python folder (e.g. python3).

Copy your source files into this folder (Note: this isn't necessary if you have python set up in the system path so that you can invoke it from anywhere).

Run the setup.bat file included in the source files. This file invokes the cx_Freeze builder.

The deployment package will be transferred to a subfolder named 'build'. From there you can just copy them wherever you like and run the pfeMAME.exe file.

That's it. You should be able to run your pfeMAME application.

NOTE: The following dependencies are required for windows and the user must take care of this themselves.

OLEAUT32.dll - C:\WINDOWS\system32\OLEAUT32.dll
USER32.dll - C:\WINDOWS\system32\USER32.dll
COMCTL32.dll - C:\WINDOWS\system32\COMCTL32.dll
SHELL32.dll - C:\WINDOWS\system32\SHELL32.dll
ole32.dll - C:\WINDOWS\system32\ole32.dll
WINMM.dll - C:\WINDOWS\system32\WINMM.dll
WSOCK32.dll - C:\WINDOWS\system32\WSOCK32.dll
COMDLG32.dll - C:\WINDOWS\system32\COMDLG32.dll
ADVAPI32.dll - C:\WINDOWS\system32\ADVAPI32.dll
WS2_32.dll - C:\WINDOWS\system32\WS2_32.dll
WINSPOOL.DRV - C:\WINDOWS\system32\WINSPOOL.DRV
GDI32.dll - C:\WINDOWS\system32\GDI32.dll
MSVCP90.dll - C:\Python27\DLLs\MSVCP90.dll
KERNEL32.dll - C:\WINDOWS\system32\KERNEL32.dll
RPCRT4.dll - C:\WINDOWS\system32\RPCRT4.dll

36 KNOWN ISSUES

36.1 KNOWN ISSUES SPECIFIC TO WINDOWS OPERATING SYSTEMS

- Startup can be a little slow (sometimes looks like nothing is happening). Have found no specific reason for it other than generic references to security software scanning it and slowing things down.

36.2 KNOWN ISSUES SPECIFIC TO LINUX OPERATING SYSTEMS

- Getting wxPython phoenix installed on Linux can be difficult at the best of times. There is a lot of information on the internet about how to do this. I have also written a procedure in this manual that always works for me.

37 Help with Installing wxPython Phoenix in Linux

To use pfeMAME in Linux, you have to be running python3 (which comes with most distributions these days) and wxPython phoenix (4.0.0 and above). The new version of wxPython can be quite difficult to install in Linux as unlike windows, you may have to build it from source.

If you are lucky, a build for your platform will already exist. Check the link below and see what's available.

https://extras.wxpython.org/wxPython4/extras/linux/gtk3

Try to find the version that most closely matches your platform and install as shown below (I'm using Ubuntu 18.04 as an example. This installed just fine on my Ubuntu 19.04 box).

pip3 install -U -f https://extras.wxpython.org/wxPython4/extras/linux/gtk3/ubuntu-18.04 wxPython

If this doesn't work, then you can try building from scratch. This comes will all sorts of dependency issues. I have listed below what I have been able to get working myself. If it doesn't work then you need to look at the errors on the screen which should guide you toward the dependency that's missing. It is also important to point out that you are going to install wxPython under python3, NOT python2.7.

The following guide is based on using a virtual environment that is the best way to perform python library installs on Linux. On newer Linux variants, some distributions actually force you to use a virtual environment. I'm also advising to use the latest Python so the instruction to download and install that is also included. NOTE: This will not replace the base Python version in your OS which would be very bad indeed.

Install the latest gtk library for python3:

      sudo apt-get install python3-wxgtk4.0

Install some critical dependencies:

      sudo apt-get install build-essential python3-dev libwebkit2gtk-4.0-dev libtiff-dev libnotify-dev freeglut3-dev libsdl1.2-dev libgstreamer-plugins-base1.0-dev

Install the Python PPA and install the latest version of Python (The example uses 3.12 but replace that with whatever version is the latest):

      sudo apt install software-properties-common

      sudo add-apt-repository ppa:deadsnakes/ppa

      sudo apt update

      sudo apt install python3.12

Install the virtual environment for the new version of Python and create a virtual environment folder (the example uses 'example-env' but name it whatever you like and put it wherever you want:

```
sudo apt-get install python3.12-venv

python3.12 -m venv example-env
```

Activate the virtual environment:

```
source example-env/bin/activate
```

Install wxPython:

```
pip install wxPython
```

Then run python using:

```
python3
```

Import wx using:

```
import wx
```

Check the version using;

```
wx.version()
```

If everything went well, you should have a valid wxPython version number. If it didn't go well then search the internet. The wxPython install log is pretty good at telling you what is missing as well so if you have any issues, please have a good read through the output of that log first.

When you are done, the following is how you disable the virtual environment:

```
deactivate
```

38 A Note About Windows XP

If you plan to build a distribution from source yourself, the highest version of python that Windows XP supports is 3.4. I have tested that this works ok with wxPython phoenix, at least for now. You will need to alter quite a bit of code manually as pfeMAME incorporates new features not found in older versions of python.