Introduction

This document provides detailed specifications for the **arcadeEIP.ini** configuration file.

An **arcadeEIP.ini** configuration file is required by the software, and must be present in its root folder. You may generate a default template of this file by running the following command at a command prompt (you can also run create_cfg.bat to create both). The files will not be created if it already exists.

```
>eip.exe -createcfg
```
[Creates **arcadeEIP.ini**]

You can also generate a special commented version by using this command:

```
>eip.exe -createcfgexample
```
[Creates **arcadeEIP_example.ini**]

This file contains abbreviated descriptions for most of the settings. Refer to this document, however, for the most complete documentation.

- **Using arcadeEIP to launch Front-Ends**
  arcadeEIP can optionally launch front-ends such as BigBox. This is particularly useful if the front-end does not support run before/after applications for control panel settings. See the folder **.\Docs\[Front_End] examples** for some examples. To do this:

  1. Ensure **arcadeEIP.ini** has a section for the front end that minimally includes the **fe_key** and **exe_full_path** settings.
  2. Also set up the **run_apps=** line (and corresponding applications) for the front-end, specifying any applications that need to run before it launches.
  3. Create a shortcut for the front end that specifies a command line of **eip.exe <fe_key>**, for example, **eip gameex** or **eip bigbox**

     When launched, arcadeEIP will…
     1. Write to a special log file while it is running called **arcadeEIP_fe.log**.
     2. Run any apps in the **run_apps=** line in **arcadeEIP.ini** for that front-end prior to launching that front-end.
     3. Run those apps again when returning to the front-end from a game.
     4. Run any apps in the **run_apps=** line of **[Front_End_os]** when exiting the front-end.

- **Using the "kill switch"**
  Use *right* Ctrl-Alt-k (or whatever sequence is defined in **arcadeEIP.ini** for the **panic_key** setting) to terminate arcadeEIP immediately. This can be useful while testing configurations that could go rogue or that you are otherwise unable to exit from. Similarly, use Ctrl-Alt-x to shutdown **marquee.exe**.

The remainder of this document contains comprehensive specifications for all sections of the **arcadeEIP.ini** file:

[General]

The **[General]** section contains settings which apply globally within arcadeEIP.
This section is not required and all its parameters are optional.

debug_mode=[fe_key or "off"]	*Optional*: Default is "off". Run in debug mode. This may be set to a valid **fe_key** value (use "os" if testing from the command-line), or to disable, set to nothing, "off", or remove the setting. Note that this can also be specified on the command-line if preferred (see Command-Line Reference Guide.pdf). Debug mode is a "dry-run" mode for testing new systems in arcadeEIP. Turning it on will do the following: a. Beep when starting and stopping (to remind you that the debug mode is turned on). b. Suppresses all file execution, however, all launch commands will be logged (in arcadeEIP.log) for review. This will enable you to examine all command-line parameters that will be used prior to actual execution. c. Automatically displays the log in either Notepad or the console window depending on where it was launched from.
status_beeps=[0 or 1]	*Optional*: Default is 0. Enables a beep upon startup and termination of **eip.exe**. This can be useful in diagnosing when arcadeEIP is being launched or terminated when being used as a launcher from another application or front-end. Note: beep-on-error is always turned on.
alpha_effect=[0 or 1]	*Optional*: Default is 1 (on). This setting activates or inactivates a fade-in effect for the start bookend screen when switching games. When turned off (0) transitions will be instant.
seed_rom=<vector>	*Optional:* Default is "mame galaga.zip". This setting specifies the rom that arcadeEIP will attempt to launch if it is run without any other rom specified on the command line or if there are no roms defined in the attract list. The setting should be provided in the form of an arcadeEIP game "vector", which is **<sys_key> <rom_file>**.
sound_device=[string]	*Optional:* Default is "Default". This setting sets the sound device that arcadeEIP will use. Ordinarily, you will want this set to "**Default**", which corresponds to the default audio device (device 1). A use case for changing this setting is If you are using MAME with port audio. In this case, because port audio takes exclusive control over the sound card assigned to MAME, you may want to have arcadeEIP use another sound card in order to support sound-effects in arcadeEIP's overlay menu while MAME is running. To obtain device names for other audio devices on your system, run the following command at a command line prompt: **>util.exe -sound**

	This will provide a list of all audio devices installed on the machine. Make sure you type the device's *name* exactly as it listed by the utility, but without the index leading number, for example: **sound_device=Speakers (2- Realtek High Definition Audio)** *The reason index numbers are not used is because these can change over time after updates, installation of new audio devices, etc.* For further details about how to use this setting, see the **sound.txt** file in the root directory of arcadeEIP. *Also note that this setting—and sound effects in general--require the 3rd party **bass.dll** library. This library is free for non-commercial use, but is not distributed with arcadeEIP due to licensing. See **sound.txt** for more information about obtaining this.*
sync_volume=[0 or 1]	*Optional:* Default is 0. This setting is only useful when **sound_device** is set to a value other than "Default". By turning this setting on, arcadeEIP will automatically try to set the volume of **sound_device** to the same value as the default audio device whenever its volume is changed (so that you don't have to manage the volume of both separately). See **sound.txt** in the root directory of arcadeEIP for more information about this setting.
show_license=[0 or 1]	*Optional:* Default is 1. This setting determines whether license text is displayed. Set **show_license=1** to show the text on the start splash screen and the Quick-Switch game picker screen, or set **show_license=0** to hide this text. Note that a license to use arcadeEIP is not required for hobbyist use, but is required for commercial use. *See the license.txt file for more information about obtaining licenses.*
hotstring_prefix=[string]	*Optional:* Default is "@!@". This setting determines the unique string that all hotstring, direct-switch filenames must start with. The default is "@!@", which means that all direct-switch filenames must start with the prefix "@!@". For example, "@!@mame_zaxxon.txt". See the readme.txt file in the \Direct folder for more information about hotstrings and direct-switch files.

EXAMPLES:

`debug_mode=os`	Debug using default front-end settings.
`debug_mode=bigbox`	Debug as though "bigbox" is the active "front-end".
`debug_mode=off`	Turn off debug mode. Could also set to zero, comment-out or remove setting.
`status_beeps=1`	Beep on startup and termination of eip.exe

`seed_rom=mame galaga.zip`	Set the seed rom to Galaga running in MAME.
`show_license=1`	Display license text in splash screens and Quick-Switch game picker screen.
`highscore_screenshots=1`	Save game screenshot whenever registering a new high score in Scorecard dialog.
`alpha_effect=0`	Turn off fade effect in bookend screens.
`hotstring_prefix=@!@`	All hotstring direct-switch files must start with the prefix “@!@”
`sound_device=Speakers (2- Realtek High Definition Audio)`	Set the secondary sound device to Realtek speakers.

[Marquee]

The **[Marquee]** section contains settings for the dynamic marquee switching feature of arcadeEIP. To use this feature, run **marquee.exe** prior to launching **eip.exe** (or turn on the autostart feature). You must set **marquee_folder** in the **[Global_Assets]** section to a valid path in order for this feature to work.
This section is optional, and not required if you do not use a dynamic marquee monitor.

active=[0 or 1]	*Optional*: Default is 0. Setting **active=0** (the default) disables the dynamic marquee feature. Setting this to 1 enables the feature.
monitor=[integer]	*Optional*: Default is 2. Set this to the monitor number that you would like your marquees to be displayed on
exit_key=[key]	*Optional:* Default is "^!x" (Ctrl-Alt-x). Use this setting to set the hotkey sequence that will exit the **marquee.exe** application.
autostart=[0 or 1]	*Optional:* Default is 0. Setting this value to 1 will cause **eip.exe** to automatically start **marquee.exe** upon startup. When this option is used, **eip.exe** will start **marquee.exe** with the **-quiet** parameter to prevent it from displaying any startup prompts.
autoexit=[0 or 1]	*Optional:* Default is 0. If this option is set to 0 (the default), **marquee.exe** will display **default.png** (if found) or perform a screen blank if **default.png** is not found, and will remain resident when **eip.exe** is exited. Setting this value to 1 will cause **eip.exe** to automatically close **marquee.exe** when it terminates.
interpolation=[0-7]	*Optional:* Default is 7. Sets the interpolation used by the GDI+ image scaler. The default value of 7 is considered optimal; however, other values might be preferred depending on performance considerations and the quality of the original images. 0 - Default interpolation mode 1 - Low-quality mode 2 - High-quality mode 3 - Bilinear interpolation. No prefiltering is done 4 - Bicubic interpolation. No prefiltering is done 5 - Nearest-neighbor interpolation 6 - High-quality, bilinear interpolation. Prefiltering is performed to ensure high-quality shrinking 7 - High-quality, bicubic interpolation. Prefiltering is performed to ensure high-quality shrinking

window_left=[integer] **window_top=[integer]** **window_height=[integer]** **window_width=[integer]**	*Optional:* Defaults are all 0. Adjusts the position and the height/width of the *window* in which the marquee image is displayed. As such, it does not directly set the size of the image itself, but rather its constraints and general positioning. Within this window, the aspect ratio of the image will be preserved and maximized. Use positive integers to increase and negative values to decrease. The values are in pixels.
show_high_scores=[0 or 1]	*Optional:* Default is 1. Determines whether high scores are shown at the bottom of the marquee when they are available. High scores may be available from the manually maintained highscores.ini file (see readme.txt in the ..\Assets folder) and/or if the third-party hi2txt feature is turned on and configured for MAME (see hi2txt in the [Systems] section of this document. Note that when high scores are displayed, marquee's will be resized to be slightly smaller (while maintaining their aspect ratio), in order to have room below for the high score.
fade_effect=[0 or 1]	*Optional:* Default is 1. Determines whether a fade transition occurs when the marquee changes. If this is set to 0, transitions will be instant. In addition, when set to 0, the poll frequency and wait frequency values will be used to govern how fast marquee transitions can occur. This can be useful under certain use case scenarios of the marquee_manager.exe (such as if you wish to change marquees while scrolling through a list, for example). Note that a restart of marquee.exe is required if this value is changed.
poll_frequency=[milliseconds]	*Optional:* Default is 80 milliseconds. Determines how often marquee.exe will check for updates. This setting is only used if fade_effect=0 (in which case poll frequency is automatically set to 1000 to allow time for the fade effect to occur). A restart of marquee.exe is required if this value is changed.

EXAMPLES:

`active=1`	Turn on the dynamic marquee feature.
`monitor=2`	Set monitor 2 as the target monitor on which marquees will be displayed.
`exit_key=^!x`	Set the hotkey that will exit the marquee application to Ctrl-Alt-x
`autostart=1`	Autostart **`marquee.exe`** when eip.exe is started.
`autoexit=1`	Close **`marquee.exe`** when eip.exe exits.
`interpolation=4`	Force interpolation mode to Bicubic
`window_left=200`	Move the marquee window 200 pixels to the left
`window_top=200`	Move the marquee window 200 pixels down
`window_height=-400`	Decrease the height of the marquee window by 400 pixels

`window_width=-400`	Decrease the width of the marquee window by 400 pixels
`fade_effect=1`	Turn on fading between the display of marquees
`poll_frequency=80`	Check for updates to the marquee every 80ms (only if fade_effect=0)

[Key_Map]

The **[Key_Map]** section contains all key maps used by arcadeEIP. When defining keys, use the key code values on the Autohotkey site for allowable values (https://www.autohotkey.com/docs/KeyList.htm) and hotkey modifiers (https://www.autohotkey.com/docs/Hotkeys.htm#Symbols). Be cautious if using modifiers since most (Shift, Alt, and Ctrl) are already used by default in the game selection menu and may introduce side-effects unless you change these defaults. *Note that all keys in this section may also be copied to individual* ***[System]*** *sections*. When this is done, the keys in those **[System]** sections will override the keys set in this section when that system is running. Also take note that many of the keys may have multiple simultaneous mappings by separating assignments with commas.

This section is required and all its parameters have defaults.

General Key – work in all modes (both in the live game and in the game selection screen).
Live Key – work only in the live game.
Menu Key – work only in the game selection screen.

panic_key=[key]	*General Key:* Default is Right-Ctrl-Alt-k. This key combination (be sure to use the right Ctrl key) will immediately kill both arcadeEIP and the currently running emulator/game when it can. It will also try to restore the mouse pointer, taskbar, and restore all resources. This is intended for use in situations where an emulator or game hangs or otherwise goes rogue (say, producing a black screen or becoming unresponsive), and other methods of exiting are not working.
lock_game_key=[key]	*Live Key:* Default is "+". This setting toggles a feature that permits the user to pause (and unpause) the attract timer. This has the effect of temporarily turning off automatic game switching to "lock" the arcade machine to the currently executing game (for this session). The locked or unlocked state is announced by both voice and by a text overlay that briefly displays the mode. If a game is locked, the timer will pause in the current session, even if the game is switched manually. Also see the **pause_lock** in the **[Auto_Switch]** section for an alternate way of invoking this same feature.
pause_key=[key]	*Live Key:* Default is "," (that's a comma). Defines the global key or sequence that will pause the currently running emulator. It is advised that you do ***not*** set this to a value that is typically used for the pause function of emulators (such as "p"). That is what the **map_pause** setting in the **[System]** section is for. Note that the **pause_key** can also have optional secondary functions tied to it. One is the lock game function (see **pause_lock**), and the other is the display of game information and image files during pause (see the **show_on_pause** setting in the [System…] section)

exit_key=[key]	*General Key:* Default is '.' (that's a period). This is the global key or sequence that will exit the currently running emulator. It is also the key value that you will typically want the exit button on your control panel configured to when using arcadeEIP. It is advised that you do ***not*** set this to a key that is typically used for the exit function of emulators (such as Esc). The actual exit commands for each emulator should be assigned to the **map_exit=** setting in each **[System]** section, which this key will call when that emulator is running. *Note that there is also a **quit_key** setting (see below), which is effectively identical in function; however, only **exit_key** may be bound to the **magic_key** function (also see below). So, if you are using a **magic_key**, but want an exit function to additionally be bound to a different key that does not exhibit the magic_key's delay (say, to exit immediately using another control panel button or external device button like an Elgato StreamDeck), then the **quit_key** could be used to define that alternative.*
menu_toggle_key=[key]<,[key]>…	*General Key:* Default is ' (that's an apostrophe). Use this key to toggle the game selection menu screen on and off. *Note: If you would like the ability to use a single button to both toggle the game selection screen and exit (or pause), consider using the **magic_key** setting.*
menu_show_key=[key]<,[key]>…	*Live Key:* Default is '='. Use this key to display the game selection screen. Unlike **menu_toggle**, this is not a toggle but only displays the menu screen if it is not currently visible (you must then use the **menu_hide_key** to turn off the menu). This key is primarily intended for use with external remote devices that need provision for discrete input.
menu_hide_key=[key]<,[key]>…	*Menu Key:* Default is '-'. Use this key to hide the game selection menu. Unlike **menu_toggle**, this is not a toggle but only hides the game selection screen if it is currently visible. This key is primarily intended for use with external remote devices that need provision for discrete input.
quit_key=[key]<,[key]>…	*General Key:* Default is '~'. This key immediately exits arcadeEIP and the currently running emulator. Its function is identical to **exit_key**, however, it can be used independent of the **magic_key** function. This can be useful if you would like a discrete exit function tied directly to a hotkey (such as on a Stream Deck), while still retaining a **magic_key** style exit function on your control panel. As a *General Key*, this key works from both the game menu and the live game.
magic_key=[key]	*General Key:* Default is '.' (that's a period, and the same default key as the **exit_key**). The "magic" key can be used to toggle the game selection screen on/off and perform a second function such as *exit* or *pause* using a single button. By default, this key is assigned to the same value as the **exit_key** (period, "."). In this default configuration, the emulator's **map_exit** function is invoked on a short press of the period key; but on a long press of at least 2 seconds

	the game selection screen is displayed instead. See the **magic_delay** and **kiosk_mode** settings in the [**Preferences**] section for information on how to further adjust this behavior.
previous_key=[key]<,[key]>…	*Live Key:* Default '[' (left bracket). Use this key to perform a direct-switch back to the previous game in the session. Note that this key only functions when the game has the focus and *will not* function when the game selection overlay is displayed. This means that if you plan to use this key, you must map it to a key that is unused by the game that is being played, otherwise there will be a conflict.
next_key=[key]<,[key]>…	*Live Key:* Default is ']' (right bracket). Use this key to perform a direct-switch forward to the next game in the Auto-Switch list. Note that this key only functions when the game has focus and will not function while the picker menu is displayed. This means that if you plan to use this key, you must map it to a key that is unused by the game that is being played, otherwise there will be a conflict.
scorecard_key=[key]<,[key]>…	*Live Key:* Default is ';' (semi-colon). Use this key to take a screenshot and enter the Scorecard Dialog. When you use this key, try to make sure your new high score is on the screen and, preferably your initials too, in order to "certify" your score. Upon entering the Scorecard Dialog you will see a thumbnail of your screenshot, and will be given instructions to enter your initials and score for registration. If you would like to re-do the screenshot, simply press the **scorecard_key** to exit the Scorecard Dialog, then press it again when the screen is more to your liking. Upon registering your score, it will be saved, along with the screenshot. Your new high score may be displayed on the marquee (if using the dynamic marquee feature with the high-score feature turned on), and the high score screenshot may be viewed during pause (see the **show_on_pause=** option in the **[Front_End_os]** section.
launch_key=[key]<,[key]>…	*Menu Key:* Default LCtrl (i.e. Left-Ctrl). While in the game selection overlay screen, use this key to launch the currently selected game. Note that by default, this is assigned to the left-ctrl key, which is the same as the player one, button one key in MAME.
list_view_key=[key]<,[key]>…	*Menu Key:* Default: 'v'. While in the game selection screen, use the **list_view_key** to switch between the list's "master" (i.e. full) view and its "picks" view showing only the games you pick from the master list. When the selected-items view is displayed, a star will appear next to the list name.
add_remove_key=[key]<,[key]>…	*Menu Key:* Default: 'Enter'. While in the game selection screen, use the **add_remove_key** to add/remove games to and from a list's "picks" view, or to/from a targeted custom list if you're in edit mode. For example, while in the master view of a system's game list, press the add/remove key, and the game will be added to its "picks" view (it will also be added to the master "Picks" list if not already there). To remove the game from a picks view, switch to it using

	the **list_view_key**, select a game in that list, then press the add/remove key (this will also remove it from the Picks list providing no other list has designated the game as a pick).
edit_mode_key=[key]<,[key]>…	*Menu Key:* Default 'e'. While in the game selection screen, use the **edit_mode_key** to target a custom list for the addition or removal of games. Custom lists are those lists defined in a **[List]** section in arcadeEIP.ini, and appear to the left of the Picks list. To target a custom list, choose the list and press the **edit_mode_key**. You may now navigate to any other list to choose games to add to the targeted list (by pressing the **add_remove_key**), or remain on the targeted list to select games to remove. Using this button on a main system list (i.e. non-custom list) has no function. Note that by default, this function is assigned to the left-shift key, which is the same as the player one, button four key in MAME.
left_key=[key]<,[key]>… **right_key=[key]<,[key]>**… **up_key=[key]<,[key]>**… **down_key=[key]<,[key]>**… **pgup_key=[key]<,[key]>**… **pgdn_key=[key]<,[key]>**… **sgup_key=[key]<,[key]>**… **sgdn_key=[key]<,[key]>**… **home_key=[key]<,[key]>**… **end_key=[key]<,[key]>**…	*Menu Keys:* Defaults: see paragraph below. While in the game selection overlay screen, use these keys to navigate the menus. <u>Defaults</u> By default, the left/right arrow keys will navigate horizontally through all visible lists, and the up/down arrow keys navigate vertically through the individual games in each menu (note that these functions can also be typically performed using joysticks or mouse devices—see **[Picker_Preferences]** section for more information about this). Page up/down, home/end, and sgup/sgdn (for sort-group up/down) provide speedier navigation through long lists. By default, page-up and page-down, in addition to being assigned to the standard PgUp and PgDn keys are also assigned to the LAlt and Space keys, which are the same as buttons 2 and 3 in MAME, and Home/End are additionally assigned to the 5 and 6 keys, which are the same as the player 1 and player 2 coin buttons. The sort-group up/down keys (sgup_key and sgdn_key) are assigned to 1 and 2 keys respectively (same as the player 1 and player 2 start buttons in MAME) and currently support paging to the previous or next leading alphanumeric character in the list. Feel free to change these assignments as you wish.
all_key=[key]<,[key]>… **favorites_key=[key]<,[key]>**… **rating_key=[key]<,[key]>**… **genre_key=[key]<,[key]>**… **year_key=[key]<,[key]>**… **publisher_key=[key]<,[key]>**… **developer_key=[key]<,[key]>**… **players_key=[key]<,[key]>**…	*Menu Keys:* Defaults: see paragraph below). While in the game selection screen, use these keys to copy or remove games that have the same assigned attribute as the game currently selected in the menu. For example, if you are in the master list of a system and the game currently selected is in the *Shooter* genre (as shown in the meta data for that game at the bottom of the screen), then pressing the **genre_key** ("g" by default) will copy all games having that genre to the "picks" view for that system (or to a custom list, if that list is currently marked as a target in edit mode). On the other hand, if a *Shooter* game is selected while in the "picks" view (or in a custom list while in edit mode), then all *Shooter* games will be removed from that list.

	Defaults This will work similarly for all assigned metadata types using the following keys: **all_key** copies/removes [a]ll games regardless of metadata values default "a" **rating_key** copies/removes all games having the same star [r]ating as selected game default "r" **favorites_key** copies/removes all games having the same [f]avorite marking as selected game default "f" **genre_key** copies/removes all games having the same [g]enre as selected game default "g" **year_key** copies/removes all games having the same release [y]ear as selected game default "y" **publisher_key** copies/removes all games having the same [p]ublisher as selected game default "p" **developer_key** copies/removes all games having the same [d]eveloper as selected game default "d" **players_key** copies/removes all games having the same [n]umber of players as selected game default "n" Keep in mind that these keys only work if the specified attribute is associated with the game. Some of these attributes are only available with LaunchBox integration (e.g. year, publisher, developer, and players). Genres can be obtained with LaunchBox integration or natively by placing a catver.ini file in the …**\Assets\[asset_name]\Meta\Info** folder.
favorite_key=[key]<,[key]>… **star_key=[key]<,[key]>…** **star5_key=[key]<,[key]>…** **star4_key=[key]<,[key]>…** **star3_key=[key]<,[key]>…** **star2_key=[key]<,[key]>…** **star1_key=[key]<,[key]>…** **star0_key=[key]<,[key]>…**	*Menu Keys:* Defaults: see paragraph below. While in the game selection screen, these keys will mark/unmark a game as a favorite or with a star rating. Importantly, once designated, you can use the **favorites_key** and/or **rating_key** (as described above) to copy all games sharing that designation to another list. Defaults: **favorite_key** [m]arks a game as a favorite or removes the marking (toggle) default "m" **star_key** marks a game with a [s]tar rating. Press multiple times to change the rating default "s" **star5_key** directly assign 5 stars to a game. default none **star4_key** directly assign 4 stars to a game. default none **star3_key** directly assign 3 stars to a game. default none **star2_key** directly assign 2 stars to a game. default none **star1_key** directly assign 1 star to a game. default none **star0_key** removes all stars from a game (0 stars). default none Note that the star0 through star5 keys are not defined by default. These are available in the event you would like to set star ratings through discrete button presses. The settings made for each game are stored in the **favorites.ini** and **ratings.ini** files located in the …**\Assets\[asset_name]\Meta\Info** folder.

EXAMPLES:

`panic_key=^!k`	Define the panic key as Ctrl-Alt-k
`pause_key=,`	Define the pause key as a comma, “,”
`exit_key=.`	Define the exit key to be a period, “.”.
`magic_key=.`	Sets the magic key to use the same key as the **`exit_key`**
`magic_key=/`	Sets the magic key to use the “/” key
`previous_key=q,[`	Sets the previous key to use either “q” or Ctrl-[
`next_key=w,]`	Sets the next key to use either “w” or Ctrl-]
`menu_toggle_key=/`	Sets the key used to toggle the game picker menu on or off
`menu_hide_key=-`	Sets the key used to turn the game picker menu off
`menu_show_key=-`	Sets the key used to turn the game picker menu on
`launch_key=LCtrl,Enter`	Sets the launch key to use either Left-Ctrl (MAME button 1) or the Enter key
`list_key=LAlt`	Sets the list toggle key to use Left-Alt (MAME button 2)
`add_remove_key=Space`	Sets the add/remove game key to use the spacebar (MAME button 3)
`edit_mode_key=LShift`	Sets the edit mode toggle key to use Left-Shift (MAME button 4)
`left_key=Left`	Sets the left key to use the Left-Arrow
`right_key=Right`	Sets the right key to use the Right-Arrow
`up_key=Up`	Sets the up key to use Up-Arrow
`down_key=Down`	Sets the down key to use Down-Arrow
`pgup_key=PgUp,1`	Sets the page up key to use PgUp or 1 (MAME player 1 start)
`pgdn_key=PgDn,2`	Sets the page down key to use PgDn or 2 (MAME player 2 start)
`lcup_key=z`	Sets the leading character group up key to z
`lcdn_key=x`	Sets the leading character group down key to x
`home_key=Home,5`	Sets the home key to use Home or 5 (MAME player 1 coin)
`end_key=End,6`	Sets the end key to use End or 6 (MAME player 2 coin)

[XInput]

The **[XInput]** section contains settings which control certain configurable attributes of XInput controllers like gamepads and fight sticks. *Note that the XInput support described here pertains to arcadeEIP functions ONLY and not to game play. To use your XInput controller in a game or emulator such as MAME, you must map your controller in that game or emulator using its native control mapping tools*. For example, if in MAME you'd like the BACK button to drop in a coin and the START button to start the game, the A button to fire, and the joystick to move your character (etc.), you must map those actions to your controller in MAME, not in arcadeEIP.

By design, XInput button mappings are generally pre-defined and not configurable by the user. However, a few can be modified for compatibility reasons. You will find options for doing that in the table following this list.

The full list of default mappings are as follows:

- **START + A:** Toggles the game selection menu on and off.
- **START + B:** Pauses and unpauses the running game.
- **START + X**: Toggles the Scorecard dialog on and off.
- **A:** Launches a selected game in the game selection menu or activates button in scorecard dialog.
- **B:** Toggles between the master and picks view of the currently selected list.
- **LB (left shoulder button):** Page-up in a game list.
- **RB (right shoulder button):** Page-down in a game list.
- **LT (left trigger):** Page-up by sort group (e.g. by first character of the game name).
- **RT (right trigger):** Page-down by sort group
- **LB + RB:** Home (jump to top of the list)
- **LT + RT:** End (jump to bottom of the list)
- **DPAD Up or Left Joystick Up:** Scroll to previous game in list.
- **DPAD Down or Left Joystick Down:** Scroll to next game in list.
- **DPAD Left or Left Joystick Left:** Scroll to previous list.
- **DPAD Right or Left Joystick Right:** Scroll to next list.

Note that these settings are also available in the **[System_]** sections if you wish to apply to specific emulators.

use_xinput=[0 or 1]	Default is 1. The **use_xinput** setting determines whether XInput support is active. It should be fine to leave this on even if you do not use any XInput controllers; however, if you know that you will not be using XInput devices (or if you experience and compatibility issues that you think might be related to XInput support), you may set this to 0 for slightly less overhead.

invert_page_buttons=[0 or 1]	Default is 0. When in the game selection menu screen, the left and right shoulder buttons (LS & RS) will page-up and page-down by screen content and the left and right triggers (LT and RT) will page-up and page-down by sort group (i.e. alphabetically). The **invert_page_buttons** setting can be used to reverse both pairs of buttons, so that if the left button paged-up before, it will now page-down, and if the right button paged-down, it will now page-up. This option is made available for preference, but also because some fight sticks have the LS and LT buttons to the right of RS and RT buttons, making it potentially desirable to reverse the mapping.
start_a=[action] ***default: "menu"*** **start_b=[action]** ***default: "pause"*** **start_x=[action]** ***default: "scorecard"***	These settings determine what action will be performed when pressing the combo buttons (i.e. START button + another button). For example, by default, pressing START and A together will toggle the game selection menu on and off. It is recommended that you keep the defaults; however, if for reasons of compatibility with particular emulator configurations or just preference you may change this to use one of the other button combinations or set any of them to "off". Setting to "off" will disable the button combination from having any effect. Supported Actions: • "**menu**": toggles the game selection menu on and off. • "**pause**": pauses and unpauses the currently running game. • "**scorecard**": toggles the Scorecard dialog on and off.

Examples:

`use_xinput=0`	Disable the use of any XInput devices
`invert_page_buttons=1`	Flip the buttons used for page-up and page-down functions.
`start_a=menu`	Toggle the game selection menu on/off when pressing START + A
`start_b=scorecard`	Toggle the Scorecard screen on/off when pressing START + B
`start_x=off`	Disable the START + Y button combination.

[Scorecard]

The **[Scorecard]** section contains settings which control aspects of the Scorecard dialog. The Scorecard dialog allows players to manually record high scores for *any* game that can be run by arcadeEIP. To use this feature, when a player obtains a high score during a game, they should press the Scorecard key (default is ';') to both take a snapshot of the game and enter the Scorecard dialog. This is usually best done immediately after entering one's initials into the game (if that option is available in the game), or at the very least, while their high score is being displayed on screen. Once in the Scorecard screen, a thumbnail of the screenshot will be shown next to the Scorecard entry dialog. The player may always exit the Scorecard dialog using ';' again to either cancel or take another screenshot. While in the Scorecard screen the player may enter their initials and score to register it by following the directions on screen. Once registered, the score is saved along with the screenshot. From this point on, the players high score may be seen (a) by calling up the Scorecard screen again at any time while the game is running, (b) by pausing the game and browsing to the high score image (see the **show_on_pause** feature), or (c) by observing the high score on the marquee if so equipped and configured.

Note that this feature can coexist with the hi2txt feature, which is able to pull high scores from nvram for display in the Scorecard and marquee for those games which support it. However, once a high score is manually registered using the Scorecard dialog, the manually entered score will be shown in lieu of any coming from hi2txt.

This section is optional since its parameters all have defaults.

take_screenshots=[0 or 1]	*Optional:* Default is 1. This setting determines whether entering the scorecard screen (using the **scorecard_key**) will take a screenshot of the active game. If it does, then this screenshot will be saved as an image file called **<rom_name>.png** in the folder designated by the **highscore_folder** setting in **[Global_Assets]** whenever the user registers a new highscore in this screen. This will overwrite any previous images for that particular game. These files provide a record of a user's high scores, and may also be shown on pause using the **show_on_pause** feature. If the user enters the Scorecard screen but does not register a new high score (they cancel or clear a high score), then no image is saved. Set **take_screenshots=** to 0 to turn this feature off.
use_watermark=[0 or 1]	*Optional:* Default is 1. Whenever a screenshot is taken when invoking the Scorecard dialog it will imprint a watermark providing the score, initials, and date/time that the score was registered. This setting allows the user to turn this water mark on when set to 1 or off when set to 0.

Examples:

`take_screenshot=0`	Turn off the screenshot feature.
`use_watermark=0`	Turn off the watermark.

[Picker_Preferences]

The **[Picker_Preferences]** section contains settings which control various behaviors and graphical attributes of the game selection menu screen.

This section is required and most of its parameters are required.

kiosk_mode=[0 or 1]	*Required.* Default is 1. The **kiosk_mode** setting controls how the **magic_key** operates when **magic_delay** is greater than zero. If **kiosk_mode=1** (the default), then the game selection menu will be displayed upon a short (i.e. ordinary) tap of the **magic_key** (*by default the period '.' key*) and the exit function will be performed on a long press (*note that only exit is supported as a secondary function in this mode*). This mode is intended/recommended for when arcadeEIP is operating stand-alone. If **kiosk_mode=0**, then the exit (or pause if so configured) function will be called on a short tap and the game selection menu will be invoked on a long press (2 seconds by default). This mode is intended/recommended for when arcadeEIP is being called from within another Front End (like BigBox) so that a quick press returns the user to the BigBox (or other front end) environment. Note that this setting (along with **magic_delay** above) may be overridden in the **[Front_End]** sections so that this behavior can dynamically vary depending on what front-end (or no front-end) is currently being used. Yes, you can have your cake and eat it too.
magic_delay=[milliseconds] *Use 0 or >500 for best results*	*Required.* Default is 2000. This setting determines how long a keypress will be needed to activate the game selection screen when/if using the **magic_key**. Setting **magic_delay=0** invokes the menu immediately; otherwise, set to a number of milliseconds (which should be at least 500—or a half-second—for best results). If the **magic_key** is set to use the same key as the **exit_key** or **pause_key**, then set **magic_delay** to a value greater than 0, such as 2000 (2 seconds). When this is done, the behavior will then be governed by the **kiosk_mode** setting. This setting (and **kiosk_mode** below) may be overridden in the [Front_End] sections.
mark_picks=[0 or 1]	*Required.* Default is 1. When set to 1 each list in the game selection screen will display a star in front of its name when toggled to display its "picks" view, and no star when it is displaying its "master" view. If

	mark_picks is set to 0, however, then this behavior will be reversed such that the master view will show a star and the "picks" view will not.
joy_vert=["all" or "off"] **joy_horiz=["all" or "off"]** **mouse_vert=["all" or "off"]** **mouse_horiz=["all" or "off"]**	*Required.* Default is "all" for joy settings and "off" for mouse settings. These settings determine whether the joysticks or mouse controllers (including trackballs/spinners if configured as a mouse) will operate as horizontal and vertical navigation controllers in the picker menu. ***Troubleshooting tip:*** *If you ever encounter an issue with menus drifting or acting "by themselves", try setting these values to "off" to see if it may be a controller calibration issue. This is especially common with the mouse settings turned on and is on my todo list as an area of investigation.*
default_list=<"picks"> or <list_key> or <sys_key> or empty	*Required.* Default is empty. This setting determines the list that the game selection screen will display at the start of each new session. Use "**picks**" to display the main "Picks" list, or choose the **list_key** of a specific **[List]** or the **sys_key** of a **[System]**. If you leave this setting blank (the default), then the list will always be set to the most recently used menu.
all_picks_label=[text] **top_picks_label=[text]**	*Required.* Defaults are both "Picks". These settings determine the text that will be displayed for the master view and picks view of the Picks list. Feel free to customize if you don't like the default text, but be aware that text should be relatively short in order to fit.
hide_systems=<sys_key><,sys_key>…	*Optional.* Default is blank. This setting will hide one or more systems from the game selection screen. Note that only **sys_key** values from the **[System]** sections are allowed here and not, for example, **list_key** values from **[Lists]**. (*To remove those, delete or comment-out their definition in the .ini file*). If you would like to hide more than one system, separate each **sys_key** with a comma. Note that hiding a system, also hides its games from other lists, including Picks, custom lists, and the Auto-Switch list. These games will reappear, however, if the **sys_key** is later removed from this setting.
clean_lists=[-1, 0, 1, or 2]	*Required.* Default is 0. Setting **clean_lists=1** will cause the displayed name of games placed in pick lists and custom lists to be stripped of any text enclosed in parenthesis () or brackets [], including those enclosures. Setting **clean_lists=2** will additionally perform this function on main system game lists. Setting **clean_lists=0** (the default) turns off this feature, but will continue to normalize any instances of double parentheses or brackets found (see below). Setting **clean_lists=-1** disables the feature completely.

	The **clean_lists** setting is global when set in this section; however, *this setting may also be copied to individual system sections* where it will override the setting in this section for that particular system. For example, if you set **clean_lists=0** in this section, and set **clean_lists=1** in the **[System_mame]** section, then the feature will be effective only for the MAME game list. If this setting is turned on, but you want some games to display parenthetical text on an exception basis, you may use double parentheses (( )) or double brackets [[ ]] in the name data source, such as in the **.meta** file or (preferably) a meta name override file. For more about these files, see the **meta_names** setting in **[Global_Assets]**.
font_type=[font name] **list_font_size=[font size integer]** **list_margin=[number of pixels]**	*Required.* Defaults are "Tahoma", 14, and 0 respectively. Use these settings to customize font and font size. The **list_margin** setting should be fine at its default setting of 0 under standard dpi (100%) settings. However, some combinations of font and non-standard Windows dpi settings may produce clipped text in the game selection menu that may be fixed by increasing this value a bit, perhaps starting with 4 and increasing/decreasing from there.
game_font_size=[font size integer] **system_font_size=[font size integer]** **game_name_ratio=[0 < value < 1]**	*Required.* Defaults are 40, 30, and 0.4 respectively. In general, only change these values if you have a strong preference for other values as certain assumptions are made in the software as to relative sizes, and so could possibly cause unintended graphical side-effects.
asset_name=[text]	*Required.* Default is "EIP_Picker". Control panel profiles can be defined for the game selection screen just as if it were another emulator. If you plan to use the **run_apps** setting in this section and want to be able to use an **[asset_name]** template in the corresponding **[Application]** section, then assign that **asset_name** here. The default **asset_name** is "**EIP_Picker**", and it is advised to keep this name unless you strongly desire another. ***Be aware*** *that while changing your control panel profiles for the picker is supported, it should only be used if you really need it since profiles can take a few seconds to load, and thus will affect how quickly the picker can be displayed. Consider making changes to the* ***[Key_Map]*** *section) to resolve any such need first if possible*, or limit control panel updates to just specific games by using the **whitelist** setting below.
run_apps=<app_key><,app_key>…	*Optional*: Default is not set. Defines one or more apps (defined in the **[Application]** sections that will be automatically run when switching to the game selection screen. Normally, only intended if you need to change control panel profiles for the picker and you are not able to use **[System]** section overrides (which are preferred—see **asset_name** setting above for additional notes).

whitelist=<vector><,vector>…	*Optional*: Default is not set. By listing game vectors in the **whitelist=** setting, the applications listed under **run_apps=** will *only* be run if one of the whitelisted games is running. It is recommended that each game is defined by its vector in the form of **[sys_key] [rom_file]** separated by a space and comma separated if there is more than one. *One example case where this can be useful is in situations where you have assigned a non-conventional profile to a joystick—such as a diagonals only profile to, say, Q*Bert and Congo Bongo. In these cases, you may have difficulty navigating the game selection menu while running these games due to the altered joystick axes. To address this, one solution is to define your joystick profiler app in the* **run_apps** *setting to have it load a conventional joystick profile, but only for when Q*Bert or Congo Bongo are running by listing these in the* **whitelist=** *setting like this:* **whitelist=mame qbert, mame congo**

EXAMPLES:

magic_delay=0	Display the game selection screen immediately
magic_delay=500	Display game selection screen (or exit, depending on **kiosk_mode** setting) after ½ second.
kiosk_mode=1	Set to invoke picker on short press and exit on long press (when **magic_delay** > 0)
joy_vert=all	All detected joysticks can be used for vertical navigation in the game selection screen.
joy_horiz=off	Joystick control for horizontal navigation in the game selection screen is disabled.
mouse_vert=all	All detected mouse controllers are enabled for vertical navigation in the game selection screen r.
mouse_horiz=all	All detected mouse controllers are enabled for horizontal navigation in the game selection screen.
default_list=picks	Make the Picks list the default list in the game selection screen.
default_list=mame	Make the MAME system the default list in the game selection screen.
default_list=autoswitch	Make the Auto-Switch list the default list in the game selection screen.
hide_systems=vic20,c64	Hide the vic20 and c64 systems (and their games) from the game selection screen.
top_rows=12	Display 12 rows of games above the Game Name bar in the game selection screen.
bottom_rows=20	Display 20 rows of games between the Game Name bar and List Name bar
font_type=Tahoma	Set the list type of the picker menu's font to Tahoma.
list_font_size=10	Set the font size used by the game lists to 10.
game_font_size=40	Set the maximum font size used in the Game Name bar (note the the font size in this control is set dynamically, so long game names may display in a smaller font size).
system_font_size=30	Set the font size used in the List Name bar.

```
game_name_ratio=0.4
asset_name=EIP_Picker
run_apps=ultramap,winipac
whitelist=mame qbert,mame congo
```

Set the width of the Game Name relative to the images on either side.
Set the name which will fill the [**asset_name**] template as “EIP_Picker”
Set the applications that will run when switching to the game selection screen (see notes).
Only run the **run_apps** applications when these games are active.

[Auto_Switch]

The **[Auto_Switch]** section provides settings for the Auto-Switch function, a central feature of arcadeEIP.
This section is required and and all its parameters are required.

folder=[relative folder path]	*Required.* Default is "Lists\Autoswitch". Set **folder=** to a relative folder path leading to a collection of vector proxy files that will define the Auto-Switch list. By default, this is set to "**\Lists\autoswitch**", which is the location where the custom list defined in **[List_autoswitch]** automatically stores its proxy files. Normally, you will not want to change this; however, if you don't plan to use the automated features below, you can technically change this to point at any other folder containing valid proxy files, including those in **\Lists\..**, **\Picks..**, and **\Direct\..**
auto=[0 or integer]	*Required.* Default is 20. When **auto** is set to a value greater than zero (for example, **auto=20**), then the Auto-Switch list will auto-populate with up to that number of the most frequently run games. Only games that have been played will be added. If set to 0, then games must be added manually. Only games in **[System]** sections having the **autoswitch=1** (opt-in) setting in them will be automatically added to the Auto-Switch list.
mute=[0 or 1]	*Required.* Default is 0. If **mute** is set to 1, then all system sound will be disabled while Auto-Switch mode is in effect. If it is set to 0, then sound always remains turned on. ***Note*** *that if this is set to 1 to mute the sound, there may be a slight delay in restoring the sound when gameplay starts (so, for example, you might not hear coin drop sounds or other initial sound effects might be clipped when the game first starts).*
delay=[seconds]	*Required.* Default is 120. Use the **delay=** setting to turn the auto-switch mode feature on or off. Set delay=0 to turn off, or a delay of at least 20 (seconds) to turn on. This setting is used to determine the number of seconds of inactivity that must occur before Auto-Switch is permitted to switch a game automatically. A rule of thumb is to always set this value higher than the expected maximum time it takes for your longest loading emulator to start-up and gameplay to start. Thus, typically, this setting should probably be no less than 30 seconds, but higher values such as 60 to 120 seconds or so are perfectly legitimate. Setting this value to zero (**delay=0**) turns off Auto-Switch completely. If used *without* the **timer** setting below, (i.e. with **timer=0**), then games will automatically switch after the designated number of seconds of inactivity. When set this way, all activity (mouse/button/controller movement) will always reset the delay counter back to zero. If your intention is to have games switch fairly frequently (say, every 2 to 5 minutes of inactivity), then leaving **timer=0** and setting **delay** to 120 to 300 seconds or so, should be fine. However, if your intention is to switch games, say, once an hour for example, then setting **delay** (to, perhaps 120) in combination with setting a **timer** to, perhaps 60 or 60C (see next setting) is recommended in order to keep any bits of stray activity from constantly pushing the scheduled switch time out another hour. Note that when **delay** is used in combination with the **timer** setting, the **delay** must be set to less time than the **timer** setting.

timer=[see text]	*Required.* Default is 5. The **timer** is essentially a scheduler, and *must* be used in combination with the **delay** setting. Unlike the **delay** setting by itself, the **timer** is *never* reset by player activity (it runs constantly); however, the **delay** setting is still needed to prevent games from switching if a user happens to be playing at the time of a scheduled auto-switch (in which case, that particular scheduled switch will simply be skipped). In general, use of the timer is advised when you want larger gaps of time between automated game switching times and/or want switching to be more regular. As a scheduler, there are several options available for the **timer**. Currently, three types of settings are supported: • **Minutes (relative):** Setting the timer to a simple integer value tells the scheduler to advance to the next game each time that number of minutes passes, starting whenever arcadeEIP was launched. This game switch will always occur unless there has been controller activity within the **delay** setting's period preceding the scheduled switch time. For example, if **timer=10** (minutes) and **delay=60** (seconds), then a game switch will occur every 10 minutes unless there is controller activity within 60 seconds of the scheduled switch time. If that happens, then game switching will be postponed until the next cycle. • **Minutes (clock-coordinated):** Setting the timer to a simple integer value followed immediately by a "C" (such as 60C) tells the scheduler to advance to the next game at the next even 60 minute interval on the clock. For example, if **timer=60C** and **delay=120** and arcadeEIP was launched at, say, 9:36AM, then the next game switch will occur at 10:00AM (providing no gamer interaction was detected within 120 seconds of that launch). The next game switch after that will then be scheduled for 11:00AM, and then 12:00PM, and so on. • **Time of day:** Setting the timer to one or more specific times a day tells the scheduler to advance to the next game at those times. This must be a time in 24-hour time and formatted as HH:MM. If multiple times will be specified, separate them with commas. For example, **timer=9:00,15:00,21:00** tells the scheduler to switch games at 9:00AM, 3:00PM, and 9:00pm every day.
random=[0 or 1]	*Required.* Default is 1. This setting determines whether the Auto-Switch list is randomized (**random=1**) or sequential (**random=0**) during playback.
pause_lock=[0,1 or # of milliseconds]	*Required.* Default is 0. This setting provides an alternate way of locking/unlocking games using the pause key (see the **lock_game_key**= setting in the **[Key_Map]** section for more information). When **pause_lock** is set to 1, the lock game function will be toggled when the user holds down the pause key for 3 seconds. When set to a number of milliseconds, this delay can be changed (for example, set to 5000 for 5 seconds). When this is set to 0 (the default), the feature is disabled.

EXAMPLES:

`folder=Lists\autoswitch`	Create Auto-Switch list from the list of proxy files in the \Lists\autoswitch subfolder.
`auto=20`	Turn on Auto-Switch, and set the maximum number of games in the list to 20.
`auto=0`	Do not add games to Auto-Switch list automatically. Manual only.
`mute=1`	Mute the system while Auto-Switch is engaged
`delay=0`	Turn off Auto-Switch.
`delay=60`	Require at least 60 seconds of inactivity before a game may switch automatically.
`timer=0`	Do not use the timer (scheduler), only use the delay setting.
`timer=10 (with delay=60)`	Switch games every 10 minutes, providing at least 60 seconds of inactivity.
`timer=60C (with delay=120)`	Switch games at the top of every hour, providing at least 120 seconds of inactivity.
`timer=15:00`	Switch games at 3pm.
`timer=9:00,15:00;21:00`	Switch games at 9am, 3pm, and 9pm.
`random=1`	Randomize the Auto-Switch list during playback.
`pause_lock=0`	Turn off the attract timer lock feature
`pause_lock=3000`	Hold the pause key for 3 seconds to pause/unpause the attract timer.

[Global_Assets]

Settings in this section provide defaults for image and textual asset paths. Settings in this section may also be copied to individual **`[System]`** sections. When this is done, those settings will override the settings in this section.

This section is not required and all of its parameters are optional (or can be set in the [System] settings instead.

***Important note about integrating with LaunchBox**:* if you use LaunchBox and specify any of the paths in this section to use assets from a LaunchBox image folder, arcadeEIP will automatically integrate with LaunchBox and also use this path information to obtain its metadata (such as game names, publisher, release date, ratings, etc.). *In this way, LaunchBox effectively becomes a scraper/content manager for arcadeEIPs graphical and metadata assets.* For example, declaring folder settings like this, will result in such integration being automatically invoked.

```
logo_folder=D:\LaunchBox\Images\[global_asset_name]\Clear Logo
marquee_folder=D:\LaunchBox\Images\[global_asset_name]\Arcade - Marquee
system_logo=D:\LaunchBox\Images\Platforms\[global_asset_name]\Clear Logo\Arcade.png
```

If you prefer to do this on a system-by-system basis rather than globally (recommended for better control/isolation), then instead of declaring these settings in this global section, you can add them instead to specific **`[System]`** sections. For example, to enable LaunchBox integration with MAME specifically, you could add these lines to the **`[System_mame]`** section (presumes the name of the MAME platform in LaunchBox is "Arcade").

```
logo_folder=D:\LaunchBox\Images\Arcade\Clear Logo
marquee_folder=D:\LaunchBox\Images\Arcade\Arcade - Marquee
system_logo=D:\LaunchBox\Images\Platforms\Arcade\Clear Logo\Arcade.png
```

More information about this topic can be found in the LaunchBox.txt file located in **`.\Docs\[Front_End] examples`** folder.

`logo_folder=[path]`	*Optional.* Default is "**`.\Assets\[global_asset_name]\Logo\Rom`**". Specifies the folder in which logo assets may be found. These will be used by bookend screens and in the game selection screen if available. Individual emulator sections do not need to define this key if it is already defined here (but may, on an exception basis if you wish). *This field supports the following templates*: **`[sys_key]`** : will replace template with the **`sys_key`** for this emulator. **`[asset_name]`** : will replace template with the **`asset_name`** for this emulator. **`[global asset_name]`** : will replace template with the **`global_sset_name`** for this emulator (most common)* **`[root_folder]`** : will replace template with the root folder of **`exe_full_path`** for this emulator.

	*The **[global_asset_name]** is usually the same as the **[asset_name]**; however, if a **[System]** section defines two **asset_names** (by separating them with a comma), then the second one will be the **global_asset_name**. Doing this can allow a system to share certain types of assets (like images or metadata) with other systems. This is what is used by the default path (shown here), which points to one of the predefined folders that comes with arcadeEIP. **logo_folder=.\Assets\[global_asset_name]\Logo\Rom** If a graphics file (either *.png or *.jpg) with a root matching the rom name is not found, then a system image file of the form **<asset_name>.png** or **jpg** will be searched for. Failing this, **.\Assets\Default\Logos\default.png** will be used.
system_logo=[file path]	*Optional:* Default is *"***.\Assets\[global_asset_name]\Logo\System\[global_asset_name].png***".* Specifies a specific system logo file that will be used when a system logo is needed or as a stand-in for the game logo if a file for the game cannot be found. The default setting looks for this file in the default folder structure named after the system's global asset name. **system_logo=.\Assets\[global_asset_name]\Logo\System\[global_asset_name].png**
marquee_folder=[path]	*Optional.* Default is "**.\Assets\[global_asset_name]\Marquee\Rom**". Currently only needed if you use dynamic marquees (see **[Marquee]** section). Specifies the folder in which marquee assets may be found. <u>Note the following special behavior:</u> When using this folder with the dynamic marquee feature, this folder will be searched for a marquee graphics file based on the root name of the rom. If that file is not found, it will next look for a file having the name of the system in the form **<asset_name>.png** (such as **mame.png**). If that file is not found, it will next use the file defined by **marquee_logo=** setting. And if not found in any of these locations, it will finally look for and use **.\Assets\Default\Marquees\default.png** if available.
system_marquee=[file path]	*Optional:* Default is "**.\Assets\[global_asset_name]\Marquee\System\[global_asset_name].png**". Specifies a specific system marquee file that will be used as a stand-in for the game marquee if a file for the game cannot be found. The default setting looks for this file in the default folder structure named after the system's global asset name. **marquee_logo=.\Assets\[global_asset_name]\Marquee\System\[global_asset_name].png**
controls_folder=[path]	*Optional.* Default is "**.\Assets\[global_asset_name]\Controls\Rom**". Specifies the folder in which control panel map image files may be found. Images in this folder may be displayed, for example, whenever pause is activated if "controls" is listed in **show_on_pause=** (see **show_on_pause** setting).

system_controls=[file path]	*Optional:* Default is "**.\Assets\[global_asset_name]\Controls\System\[global_asset_name].png**". Specifies a specific system controls file that will be used as a stand-in for the system control image if a file for the game cannot be found. The default setting looks for this file in the default folder structure named after the system's global asset name. **system_controls=.\Assets\[global_asset_name]\Controls\System\[global_asset_name].png**
highscore_folder=[path]	*Optional.* Default is "**.\Assets\[global_asset_name]\HighScore\Rom**". Specifies the folder in which high score screenshots will be saved (see **highscore_screenshots** setting). Images in this folder may be displayed, for example, whenever pause is activated if "controls" is listed in **show_on_pause=** (see **show_on_pause** setting).
meta_names=[full file path]	*Optional.* Default is "**.\Assets\[global_asset_name]\Meta\Names\.meta**". Specifies full path and filename for the meta name files (not needed for BigBox since arcadeEIP is able to read its xml metadata files directly). *Note that this setting should not be needed if you are integrating with LaunchBox since the meta data in that case, will be obtained from the LaunchBox database instead.* For example: **D:\Assets\[global_asset_name]\Meta\Names\.meta** Meta name files map rom names to human names. There are two ways of doing this. 1. A single file placed in the **meta_names** designated folder (by default named ".meta") may be created that contains all the name mappings for the emulator. The format of each line in the file must be: **unquoted rom name "double-quoted rom description"** For MAME, such a file can be created easily by typing this command in the mame folder: **mame.exe -listfull > .meta** This file can then be copied to the **meta_names** designated folder. Note that the GameEx emulator uses this format for its map files. Thus, these may also be used if you use that front end. To do this, it is recommended that you copy the **meta_names=** key to the appropriate **[System]** section and hardcode the path to the specific map file you want to use for that emulator.

	2. Name mappings may also be created in individual files. In such cases, the filename should be the rom name, but with the extension .txt. The contents of the text file should be simply the game description. Note that both a .meta file and individual files are used, the individual files will override the contents of the .meta file for that game. This can be useful. For example, if the .meta file contains this line for Battlezone: **`bzone "Battle Zone (rev 2)"`** …and you would prefer for the name to be "Battlezone", you can create a file called **`bzone.txt`** containing the text "Battlezone" and place it in the **`meta_names`** folder. This will cause the name for just that game to be overridden. Note that if the **`meta_names`** key is defined in both the **`[Global_Assets]`** section and in a **`[System]`** section, the path in the **`[System]`** section will override this one; however, if both paths are defined, you may place individual override files in either folder. *This field supports the following templates*: **`[sys_key]`** : will replace template with the **`sys_key`** for this emulator. **`[asset_name]`** : will replace template with the **`asset_name`** for this emulator. **`[global asset_name]`** : will replace template with the **`global_sset_name`** for this emulator (most common)* **`[root_folder]`** : will replace template with the root folder of **`exe_full_path`** to the emulator.

EXAMPLES:

```
logo_folder=D:\Assets\[asset_name]\Logos
system_logo=D:\Assets\[asset_name]\System_Logo\
                                    system_logo.png
```

A path to game logo png files using the **`[asset_name]`** template. Used by bookend screens.
A path to system logo png files using the **`[asset_name]`** template. Used by splash screens.

```
marquee_folder=D:\Assets\[asset_name\Marquees
Controls_folder=D:\Assets\[asset_name]\Controls
meta_names= D:\Assets\[asset_name]\Meta\Names\.meta
```

A path to marquee png files using the **`[asset_name]`** template. For dynamic marquees.
A path to control panel map image files. The default used for the image shown on pause.
A path to a name lookup file mapping rom names to human names.

[Front_End_os]

Settings in this section pertain to the default front-end, which must always be the native operating system (i.e. Windows).
This section is required and must minimally contain the definition, `fe_key=os`
(Use other front end sections named using the format, `[Front_End_<fe_key>]` to specify additional front-ends (see next section)).

`fe_key=os`	***Required***. Must contain the default definition: `fe_key=os`.
`asset_name=[text]`	*Optional*. Default is "windows". Sets the string which will be inserted into the `[asset_name]` template for the *os* front end. "windows" is recommended, but can be set to another value if desired.
`proper_name=[text]`	*Optional:* Default is "Windows". Sets the string that will be inserted into the `[proper_name]` template for this front-end. This is the "human" name of the front-end and may contain spaces.
`start_screen=[sec]<,[color]>`	*Optional*. Default is 0. Set to -1 to turn off. Displays the startup bookend (loading) screen before the emulator runs for a minimum of `[sec]` seconds. Actual display time may exceed the specified time if `run_apps` are defined and they take longer than `[sec]` seconds to run. Set to 0 for minimum display time. `[color]` defines the background color. May be an HTML color (like "Navy") or a hex RGB value (like 0x808080). See https://www.autohotkey.com/docs/commands/Progress.htm#colors
`exit_screen=[sec]<,[color]>`	*Optional*. Default is 1. Set to -1 to turn off. Displays the exit bookend screen after the emulator exits. Takes same arguments as `start_screen`
`hide_cursor=[0 or 1]`	*Optional*. Default is 0. Hides (1) or shows (0) cursor for this front-end. It is recommended to keep this set to 0 when testing new emulators to avoid loss-of-cursor in the event of an application crash.
`hide_taskbar=[0 or 1]`	*Optional*. Default is 0. Hides (1) or shows (0) the Windows taskbar for this front-end. Default is 0. It is recommended to keep this set to 0 when testing new emulators to avoid loss-of-cursor in the event of an application crash. Note that this setting should only be needed when running emulators that use the "nohide" option tag.
`run_apps=<app_key><,app_key>…`	*Optional.* Default is not set. Defines one or more apps (defined in the `[Application]` sections) that will be run whenever returning to the os. In addition, the following prefix is supported. "+" : Applications prefixed with a "+" will be run before any apps called in an emulator section.

show_on_pause=[asset type]	*Optional:* Default is "**native,highscore,controls,info**". Specifies the pages (in order) that will be displayed when the pause key is pressed. In this case, the "native" page will always be shown first. Scrolling to the right will then display the "highscore", "controls", and "info" pages, while scrolling to the left will display the "info", "controls" and "highscore" pages. Continuing to scroll in a particular direction will wrap back to the beginning or end in a continuous loop. If no image or data is available for a specific page, system, or game, then that page will simply be skipped. Pages may be re-ordered in any way the user wishes. This setting may also be left blank if you do not wish any such pages to be displayed. Available asset types include: • **native** : This "page" is simply a transparent window that will show the emulator's native pause screen. • **highscore**: This page will show any high score image associated with the current game. *Highscore pages for games are created automatically in the* ***.\Assets\<system>\Highscore\Rom*** *folder through use of the Scorecard dialog. To use the Scorecard dialog press the ";" (semi-colon) key (or any button calling ';' as a hotkey) during gameplay to take a snapshot of the game screen while it is showing your highscore (try to capture your initials as well if possible); then follow the direction in the dialog shown on screen to enter your initials and score, and to save (or update) the highscore snapshot. See the scorecard_key setting in the* ***[Key_Map]*** *section for more information.* • **info**: This page will display information and history for the current game if available. Currently, this is only available for MAME. While this is called a "page", the *info* may in fact span several pages. *The "info" option is designed to combine and display game data from the MAME* ***history.xml*** *and/or* ***mameinfo.dat*** *files. For this option to work, there must be a* ***mameinfo_folder=*** *in the* ***[System]*** *section of each system that uses MAME, or which uses MAME style rom names (like Supermodel, etc.). This setting must point to a folder that contains the MAME* ***history.xml*** *and/or* ***mameinfo.dat*** *files (by default, arcadeEIP ships with compatible builds of these files in the* ***.\Assets\MAME\Meta\Info*** *folder). If both are provided, data from both will be merged and displayed together. Note that you must use the XML version of the history file since the .dat format is not supported. For more information about this setting, see* ***mameinfo_folder=*** *entry in the* ***[Systems]*** *section below.* • [***image-asset-type***]: This is the root name of an image type specified in the **[Global_Assets]** section of arcadeEIP.ini. For example, "***controls***", "***logos***", "***marquees***", etc. Custom types *are* permitted. Currently, only still images of common types (like *.jpg, *.png, etc.), and not videos, are supported.

	To specify a particular asset type, use the root name of one of your asset types. For example, if you have a ***logo_folder=*** *line specified in* ***[Global_Assets]*** *or in one or more* ***[System]*** *sections, then the asset type would be "****logo****", and thus you may add this to the list, such as* ***show_on_pause=controls,logo,info****.* *You may also use custom asset types by adding new keys to the* ***[Global_Assets]*** *section (or* ***[System]*** *sections) following the form of existing entries. For example, if you created this key…* ***flyer_folder=*** *…you would be able to specify,* ***show_on_pause=flyer*** *to display flyer brochures on pause.* *Like all other settings in this section, this setting may be overridden in the individual* ***[System]*** *sections to display different media on pause for different emulators.*
magic_delay=[milliseconds] *Use 0 or >500 for best results*	*Optional.* Default is not set. This setting (and **kiosk_mode** below) may be used to override the setting in the **[Picker_Preference]** sections.
kiosk_mode=[0 or 1]	*Optional.* Default is not set. This setting (and **magic_delay** above) may be used to override the setting in the [Picker_Preference] sections.

EXAMPLES:

fe_key=os	Literal and required
asset_name=windows	Specifies "windows" as the string which will be inserted into the **[asset_name]** template.
proper_name=Windows	Specifies "Windows" as the "human" name of this front-end.
start_screen=3	Displays the start screen for a minimum of 3 seconds
start_screen=0	Displays the start screen for the minimum amount of time it takes for apps to run.
start_screen=3,Navy	Displays the start screen for 3 seconds with a Navy blue background
start_screen=4,0x808080	Displays the start screen for 4 seconds with a gray background
exit_screen=3	Displays the exit screen for a minimum of 3 seconds
hide_cursor=1	Hides the cursor while bookend screens are being displayed.
run_apps=winipac,ultramap	Runs the applications with **app_key=winipac** and **app_key=ultramap** in order, whenever returning to the os. If the application uses the **[asset_name]** template, it will be filled in by this section's **asset_name** (for example, "Windows").
run_apps=+cdcreate,winipac	Runs the **cdcreate** application on launch (prior to any applications specified by the emulator). Runs the **winipac** application upon return to the os.

`show_on_pause=controls,info` Specifies that a control panel image and MAME info should be displayed on pause.

[Front_End_<fe_key>]

Use additional sections to define *additional* front-ends. **Settings are the same as listed above in the `[Front_End]` section above, with the exceptions listed below.**

`fe_key=[text]`	***Required***. Must contain a *unique* id for this front-end (unique among all **`fe_key`**, **`app_key`**, **`list_key`**, and **`sys_key`** values in this configuration). Should match the name given in the section header. Do not use spaces, and only use characters acceptable in a filename.
`exe_full_path=[exe_path]`	***Required***. Full path to the front-end's executable.
`proper_name=[text]`	*Optional:* Sets the string that will be inserted into the **`[proper_name]`** template for this front-end. This is the "human" name of the front-end and may contain spaces.
`window_name=[text]`	*Optional:* This should be whatever text appears in the front-end's title bar. If the front-end's title bar is hidden, you can usually discover this text by doing Ctrl-Alt to enter the task switcher and observing the text over the thumbnail image representing the emulator. Be sure to enter the text exactly (including correct case). This text will be used when arcadeEIP exits in order to return focus to the front-end that launched it. Some front-ends may not require this, but some (like Big Box) do.
`run_apps=<app_key><,app_key>`…	*Optional.* Defines one or more apps (defined in the **`[Application_n]`** sections) that will be run (a) upon front-end startup and (b) whenever returning to this front-end from an emulator. If an application uses the **`[asset_name]`** template, it will be filled in with the **`asset_name`** defined in this section. *In addition, the following prefixes are supported.* "+" : Applications prefixed with a "+" will be run *only* upon initial front-end startup. "-" : Applications prefixed with a "-" will be run *only* upon front-end shutdown.
*Additional settings defined in the default **`[Front_End_os]`** section may be used in these additional sections as well.*	

EXAMPLES:

`fe_key=gameex`	Specifies a key for the GameEx emulator.
`asset_name=GameEx`	Specifies "GameEx" as the string which will be inserted into the **`[asset_name]`** template.
`proper_name=GameEx`	Specifies "GameEx" as the "human" name of the front-end.

`exe_full_path=C:\GameEx\GamEx.exe`	Specifies the path to the front end.
`run_apps=joytokey,winipac,ultramap`	Specifies three controller configuration apps that should be run (in order) when the front-end is initially launched (by arcadeEIP), and when returning to the front-end after exiting an emulator.
`run_apps=+cdcreate,winipac,-cdremove`	Runs the **`cdcreate`** application only upon initial launch of the front-end. Runs the **`winipac`** application upon initial launch of the front-end, *and* whenever returning to the front-end from an emulator. Runs the **`cdremove`** application only upon front-end shutdown.
`window_name=LaunchBox Big Box`	The title bar text for Big Box. This enables arcadeEIP to return focus to Big Box after exiting.

[Integration]

This section provides settings that affect integration with other 3rd party software**. Presently, these settings are only compatible with LaunchBox integration.**

Note that importing favorites or ratings is not a permanent operation in that imported settings are merely read into memory upon startup and not actually saved in arcadeEIP; thus, these can be removed by simply turning the setting off. If you ever wish to reset/remove the favorites or ratings that have been set in arcadeEIP, simply delete the appropriate in file. That would be **..\Assets\<system>\Settings\favorites.ini** for favorites or **..\Assets\<system>\Settings\ratings.ini** for ratings (or any specific keys in those files). Finally, note that any change to favorites or ratings made in arcadeEIP remain in arcadeEIP and are not transferred back into LaunchBox (arcadeEIP never writes to the LaunchBox databases).

import_favorites=[0 or 1]	*Optional.* Default is 0. When integrating with LaunchBox (see notes in the **[Global_Assets]** section as well as **..\Docs\[Front_End] examples\LaunchBox.txt**), setting **import_favorities=1** will cause arcadeEIP to use the Favorite settings from LaunchBox for all games that have not already been marked (or unmarked) as a Favorite in arcadeEIP (i.e. Favorite settings made in arcadeEIP shall always take priority over those being imported). When set to 0 (the default), favorites from LaunchBox will not be used at all.
import_user_ratings=[0 or 1]	*Optional.* Default is 0. When integrating with LaunchBox (see **..\Docs\[Front_End] examples\LaunchBox.txt**), setting **import_user_ratings=1** will cause arcadeEIP to use the manually entered user Rating values from LaunchBox for all games that have not already been given a rating in arcadeEIP (i.e. Rating settings made in arcadeEIP shall always take priority over those being imported). When set to 0 (the default), user ratings from LaunchBox will not be used at all.
import_community_ratings=[0 or 1]	*Optional.* Default is 0. When integrating with LaunchBox (see **..\Docs\[Front_End] examples\LaunchBox.txt**), setting **import_community_ratings=1** will cause arcadeEIP to use the community Rating values from LaunchBox for all games that have not already been given a rating in arcadeEIP or a manually entered rating in LaunchBox (i.e. Rating settings made in arcadeEIP shall always take priority over those being imported). When set to 0 (the default), community ratings from LaunchBox will not be used at all.

EXAMPLES:

import_favorites=1	Import favorites from LaunchBox (requires integration to be active).

`import_ratings=1` Import ratings from LaunchBox (requires integration to be active)..

[List_<*list_key*>]

Each **`[List_<sys_key>]`** section defines a custom list. Replace <list_***key***> with the unique **`list_key`** provided in the section. **These sections are optional, although the default Auto-Switch is recommended for core functionality.**

`list_key=[text]`	***Required***. Must contain a *unique* id for this list (unique among all **`fe_key`**, **`app_key`**, **`list_key`**, and **`sys_key`** values in this configuration). Should match the name given in the section header. Do not use spaces, and only use characters acceptable in a filename.
`proper_name=[text]`	*Optional:* Sets the string that will be inserted into the **`[proper_name]`** template for this list. This is the "human" name of the list and may contain spaces.
`sort_prefix=[text]`	*Optional:* Defines a string that will prefix the **`proper_name`** when sorting the custom lists for display, such as in the menu. This value is never shown to the user, but is only used for internal sorting. For example, say, four custom lists are defined with the proper names "Auto-Switch", "Golden Age", "Miscellaneous", and "Shooters". By default, these will be listed in the menu alphabetically as shown by proper name. However, suppose you would prefer the order to be: "Auto-Switch", "Golden Age", "Shooters", and "Miscellaneous". To do this, you could add **`sort_prefix=t`** (or any text that would follow "Shooters" alphabetically) to the section defining the *Miscellaneous* list.

EXAMPLES:

`list_key=golden`	Create a list with a key called "golden"
`proper_name=Golden Age`	Give the list the proper name "Golden Age"
`sort_prefix=AA`	Prefixes the proper name with "AA", likely putting it first in the menu.

[System_<*sys_key*>]

Each **[System_<sys_key>]** section defines an individual emulated system. Replace **<*sys_key*>** with the unique **sys_key** provided in the section.

At least one System section is required and must minimally contain the settings designated as "Required" below. If you use MAME, it is recommended to configure this as your first system.

General Note about fullscreen vis. windowed modes*: Because the game picker menu is an overlay, the exclusive fullscreen mode that some emulators use by default will prevent the overlay from being displayed. Fortunately, most emulators these days offer an alternative fullscreen windowed mode that will work. So, when configuring your emulators, make sure that you set them up to use their windowed mode rather than exclusive fullscreen for best results. All systems having configuration examples documented in the* **\[System] examples** *folder of the distribution possess compatible fullscreen windowed modes. See that documentation for more information.*

sys_key=[text]	***Required***. Must contain a *unique* id for this system (unique among all **fe_key**, **app_key**, **list_key**, and **sys_key** values in this configuration). Should match the name given in the section header. Do not use spaces, and only use characters acceptable in a filename.
asset_name=[text]<,text>	*Optional but strongly recommended.* Sets the string that will be inserted into the **[asset_name]** and **[global_asset_name]** templates for this emulator. Only use characters acceptable in a filename. Spaces are allowed. *Note:* When specifying a single value (the most frequent case), this will fill-in both **[asset_name]** and **[global_asset_name]** template fields with that same value. However, if a second value (after a comma) is specified, for example: **asset_name=naomi-digital,naomi** then the first value will be used to fill in **[asset_name]** and the second value will be used to fill in **[global_asset_name]**. This permits more nuanced resource management in cases, for example, where one wants to share image assets with another system (say, in this case, one that defines **asset_name=naomi-analog,naomi**), while retaining the ability to keep other assets (such as control panel configuration profiles) separate. *This key additionally supports the following template*: **[sys_key]** : will replace template with the **sys_key** for this emulator.

proper_name=[text]	*Optional but highly recommended:* Sets the string that will be inserted into the **[proper_name]** template for this system. This is the "human" name of the system and may contain spaces. Its most obvious use is as the system's name in the game selection menu.
sort_prefix=[text]	*Optional:* Defines a string that will prefix the **proper_name** when sorting the systems for display, such as in the menu. This value is never shown to the user, but is only used for internal sorting. For example, say, four systems are defined with the proper names "Daphne", "MAME", "Naomi", and "Pinball". By default, these will be listed in the menu alphabetically as shown by their proper names. However, suppose you would prefer the order to be: "MAME", "Daphne", "Naomi", and "Pinball". To do this, you could add **sort_prefix=AA** to the MAME section to force it to be considered the first one alphabetically.
autoswitch=[0 or 1]	*Optional:* Determines whether this system participates in the Auto-Switch feature. If this setting is 0 (or the setting is omitted), then games from this system will *not* be automatically registered for inclusion in auto-switching. If **autoswitch** is set to 1, then the system *will* participate. It is generally recommended that this setting should only be set to 1 in systems that have games that will ordinarily start in an immediately playable attract mode, such as true arcade games, for example.
exe_full_path=[exe path<,min><,nohide><,winset_A>…	***Required***. Full path to the emulator's executable including the executable name. You may also use this option to specify one or more comma separated run option tags. Option tags include the following: • **min**: runs the emulator minimized. This is helpful when running emulators like MAME which spawn a console window when starting up. This option helps to hide this. • **nohide**: Try using this tag if you cannot see the emulator when it runs (i.e black screen is displayed). This is helpful for emulators like Visual Pinball, WinVICE, and others. • **winset_A, winset_B, winset_C, winset_D**: Emulators sometimes exhibit anomalies in their launch behavior. For example, they may display window borders, title bars, etc. The winset_*X* methods are special routines to try correcting these. Each was developed with a specific emulator in mind; however, they can also be used generically. There are four that exist currently. These cannot be combined. Note that **window_name** (see next setting) must be correctly set in the section in order for A-C to work.

	Variants: **winset_A**: Attempts to hide window title, menu, and borders. Designed for Altirra. **winset_B**: Designed for Supermodel. Performs winset_A functions while also centering screen. **winset_C**: Designed for Demul. Combines winset_A and winset_B while also including a Demul-specific delay. **winset_D**: Designed Visual Pinball. Moves the mouse during startup to prevent taskbar pop-up. *This key additionally supports the same templates as* **`param_list`** *(with the exception of* **`[root_folder]`***).*
`window_name=<window text> or <"process">`	***Optional, but recommended*** ***Technical advisory*****:** Read this section carefully before setting **`window_name`** since setting this to a wrong value *can cause rogue behavior that may cause loss of control or lock-up conditions such that you may need to reboot your computer.* The **`window_name`** setting is required in order to use a **`winset`** run options (see **`exe_full_path`** above). For other systems, this setting is optional, but when used will cause arcadeEIP to keep emulator focus more aggressively, which can make software more stable under conditions where users might be, say, mashing buttons while emulators are switching, hitting alt-tab at a bad time, etc. Set this to either: 1) The window title text of the *main* GUI window for emulators that spawn multiple windows when they are launched. It is ok to use partial text *as long as it is unique, always there, and would not be likely part of another window's text* (such as another spawned window, or Windows Explorer, for example). If, as in MAME, the text changes based on the loaded rom, it is ok to use the **`[rom_name]`** or **`[rom_file]`** templates. For example, for MAME, setting **`window_name=[[rom_name]]`** will work because MAME always puts the rom name (without extension) in brackets in the Window title of its main GUI window. OR 2) Set **`window_name=process`** if you are certain the emulator only spawns a single window. In this case, the window will be discovered automatically by arcadeEIP from the process id. If you don't set **`window_name`**, arcadeEIP should still work very well; thus, with the exception of emulators needing a **`winset`** setting, this setting is not required. However, if an emulator does lose focus (resulting in black screen or unresponsiveness) you will need to recover manually using alt-tab, and while the emulator is out of focus, keys such as exit may not work.

	The ini sample library provides example values of this setting for many emulators, so if you do use this setting, use those examples for guidance.
search_path=[path]*.[* or ext]<,ext>…<||additional search paths>	***Strongly Advised, but optional if using arcadeEIP strictly as a launcher.*** Adding a **search_path** enables running a rom without having to specify a full path; it also enables partial rom name matching which can be handy at the keyboard, especially if rom names are long or decorated. And, perhaps most importantly, it makes it possible for the game selection screen to create its game lists (if you won't be using these features, and are only using arcadeEIP to launch games from a front-end using a full path, then you don't necessarily need a **search_path**. The **search_path** setting is compatible with the **<change_root>** rule. When specifying a **search_path**, use specific wildcards for the file names (e.g. ***.rom** or ***.bin,zip** vs. ***.***) to better constrain the selections, though either form is allowed. For example, using *.* while also using proxy files in the same folder could lead to spurious matching. To match on more than one file extension, separate them using commas. If roms are found in separate folders, you may specify these by using a double pipe "||" to separate multiple paths. For example: **search_path=D:\Emulators\Demul\Demul_digital\roms*.zip||D:\Emulators\Demul\Demul_analog\roms*.zip,*.bin** Adding a **search_path** also enables use of the **-list, -find, and -findall** command-line arguments. See the "Command-Line Reference.pdf" document for more about these commands. This only works in a console window, but will allow you to list all the roms in the search spec for an emulator using the form **util.exe [sys_key] -list** Keep in mind that partial matching can be inexact.
mameinfo_folder=[path]	*Optional.* Default locatin for this file is "**.\Assets\MAME\Meta\Info**". Specifies a folder that contains the files **history.xml** and/or **mameinfo.dat**. This line should only be specified in **[System]** sections that use MAME as the emulator, or which use roms that are named using the same convention as MAME (such as Supermodel, etc.). This line permits use of the "info" option in the **show_on_pause=** setting described in the **[Front_End]** section above. Note that only the XML version of the history file is supported, not the .dat form.
logo_folder=[path]	*Optional.* Specifies the folder in which logo assets may be found. These will be used by bookend screens if available. This key is not needed if there is already an appropriately configured **logo_folder** definition in the **[Global_Assets]** section.

	See the [**Global_Assets**] section for more information about the use of this and other asset folder settings, **including how to use this setting to invoke integration with LaunchBox.**
system_logo=[file path]	*Optional.* Specifies path and filename to a specific system logo file. Not needed if there is already an appropriately configured **system_logo** definition in the **[Global_Assets]** section. See the **system_logo** setting description above in the [**Global_Assets**] section for more information.
marquee_folder=[path]	*Optional.* Specifies the folder in which marquee assets may be found. These will be used by the dynamic marquee monitor feature. This key is not needed if there is already an appropriately configured **marquee_folder** definition in the **[Global_Assets]** section. See the **marquee_folder** setting description above in the [**Global_Assets**] section for more information.
system_marquee=[file path]	*Optional.* Specifies path and filename to a specific system marquee file. Not needed if there is already an appropriately configured **system_marquee** definition in the **[Global_Assets]** section. See the **system_marquee** setting description above in the [**Global_Assets**] section for more information.
controls_folder=[file path]	*Optional.* Specifies the folder in which control panel map image files may be found. If a file containing such an image is found for a game, then that image will be displayed whenever pause is activated. See the **controls_folder** setting description above in the [**Global_Assets**] section for more information.
meta_names=[full file path]	*Optional.* Specifies full path and filename to the meta names resources for mapping rom names to human names. Not needed if there is already an appropriately configured **meta_names** definition in the **[Global_Assets]** section. Also not needed for BigBox since arcadeEIP is able to read its xml metadata files. See the **meta_names** setting description above in the [**Global_Assets**] section for more information.
show_on_pause=[asset type]	*Optional:* Specifies what kind of image will be displayed when the pause key is pressed for this particular system. Not needed if there is already an appropriately configured **show_on_pause** definition in a **[Front_End]** section. See the **show_on_pause** setting description above in the [**Front_End_os**] section for more information.

param_list=[parameter_list]	*Optional.* Specifies the default runtime parameters to be used by this emulator. *This field supports the following templates. Note that it is generally advised to place quotes around the template name if the value might include spaces, although requirements may vary by emulator.* **[rom]** : Replace template with **rom_full_path** from the command line if available; if full path not available, then **rom_file** will be used; if that is not available, uses **rom_name**; otherwise blank. **[rom_full_path]** : Replace template with full path to the rom (including rom file) if available, blank if not. **[rom_file]** : Replace template with the **rom_file** (rom name plus extension) if available, blank if not. **[rom_name]** : Replace template with the **rom_name.** **[rom_folder]** : Replace template with the **rom_folder** if available, blank if not. **[rom_folder_parent]** : Replace template with the folder one level higher than the rom folder. **[sys_key]** : Replace template with the **sys_key** for this emulator. **[asset_name]** : Replace template with the **asset_name** for this emulator **[root_folder]** : Replace template with the root folder of **exe_full_path** for this emulator <u>Passthrough Parameters</u> This field also supports passthrough parameters using the form **<param,number_of_args>**. Use this if the front-end passes an argument on the command-line that you wish to add to your parameter list only if the front-end supplies it. For example, suppose the front-end places the argument **-video ddraw** into the command line when connected to an RDP session. To allow this to pass through, add the value **<-video,1>** to the parameter list. This instructs eip.exe to look for a parameter called **-video** on the command line, and if found, include it and its one argument (**ddraw** in this case) in the parameter list that it uses to launch the game. If the parameter is standalone and doesn't support arguments (like **-autosave**), specify 0 as the *number_of_arguments*.
run_apps=<app_key><,app_key>…	*Optional.* Defines one or more applications (defined in the **[Application]** sections that will be run whenever starting this emulator. This is used for configuring control panels, running Joytokey, mounting CDs, etc. *The following prefix is supported.* "-" : Applications prefixed with a "-" (minus sign) are run upon emulator exit rather than start. When used, note that the **[asset_name]** template will always be filled in with the **asset_name** of the *front-end*, not the emulator. Also note that it is recommended that applications that should run on emulator exit that are *specific to a particular front end* (like a control panel configuration) should be added to the front-end's **run_app** section rather than the emulator's.

	In addition, the following built-in functions are supported which may be placed in-line with apps. **`delay([milliseconds])`** – inserts a delay for specified number of milliseconds between apps **`send([message],<[wait delay]>`** – async delay for number of milliseconds, then sends message **`killwin([window title text])`** – closes a named window (partial text ok) **`minimize([window title text])`** – minimizes a named window (partial text ok) **`maximize([window title text])`** – maximizes a named window (partial text ok) **`hidewin([window title text])`** – attempts to make named window transparent (partial text ok) **`showwin([window title text])`** – restores a named window from being transparent (partial text ok)
`pause_apps=<app_key><,app_key>`…	*Optional.* Defines one or more applications (defined in the **`[Application]`** sections that will be run whenever pausing or unpausing this emulator. The same options are available as **`run_apps=`** above, however, the "-" prefix in this setting is used to designate that the application is to be run on unpause.
`map_exit=[mapping]`	*Optional.* Creating a **`map_exit`** command is an essential part of a system's configuration since both the **`exit_key`** operation and the ability to switch games depends heavily on it. There are three major ways to specify an exit sequence. 1. Send whatever keystrokes the emulator defines for its normal exit sequence (e.g. Esc, Ctrl-x, Alt-F4, etc.), and/or, 2. Send close messages to the emulator's window(s); and/or 3. Terminate the emulator process. Since emulators vary widely in how (or even if) they are able to see virtual keystrokes. Different methods might need to be attempted. This can be complicated further if the emulator spawns multiple windows and processes. Of course, the most turn-key way to find the exit sequence for your emulator is to see if one is already documented in the **`\Docs\[System] examples`** folder. If not, you should review AutoHotKey documentation (https://www.autohotkey.com/docs/commands/Send.htm) along with reading the following specifications for this setting: Specifications: The **`map_exit`** value consists of either one or two strings. If two, then they are separated by a pipe ("|") character. The part to the left of the pipe is always executed first, and the part to the right (if defined) is executed second. If tied to a physical key, the left sequence is executed when the button is pushed down, and the right sequence when the button is released. The options, and roughly, the order in which these operations will be carried out is the same on both sides. **`map_exit=<{CloseTop}><{CloseActive}><{WinClose}><{EndProc}><{Event}><KeySequence>|<{CloseTop}><{CloseActive}><{WinClose}><{EndProc}><{Event}><KeySequence>`**

The following descriptions apply to these tags and general usage. Note that combining tags is generally fine; however, always use the *absolute minimum* needed to get the job done--most exit sequences are really quite simple.

- A simple "KeySequence" will have no tags and consists of just one or two virtual keystrokes to send using the syntax specified by the AHK documentation. For some sequences, note, however, that the additional modifiers **`{Normal}`** and **`{Blind}`** may be required.

- To define a KeySequence that emulates a key-toggle action (some emulators might require this), use a "down" or "DownR" modifier after the key to the left of the pipe ("|"), and an "up" modifier after the key specified to the right.

- A KeySequence that requires SendEvent style messages must be preceded with the tag **`{Event}`**. These keystroke events can use special timing. By default **`{Event}`** specifies a delay of -1 (i.e. no delay) and press duration of 110ms. However, you may add comma separated values to change this if needed. For example, **`{Event,-1,-1}`** sets both delay and duration to -1. For example, **`{Event}{Esc}`**. Do not use **`{Blind}`** or **`{Normal}`** with **`{Event}`** type sequences.

- Use the **`{CloseTop}`** tag by itself or in combination with other tags to send a close message to the topmost fullscreen window (which is not always the active window).

- Use the **`{CloseActive}`** tag by itself or in combination with other tags to send a close message to the current active window.

- Use the **`{WinClose}`** tag by itself or in combination with other tags to close the emulator window. By default, **`{WinClose}`** will target the **`window_name`** key in the section. Otherwise, follow **`{WinClose}`** by one or more (comma-separated) window titles to shutdown. For example, **`{WinClose,Virtual Pinball,VPinball}`**. Partial text is ok in these instances, but make sure it is unique.

- Use the **`{EndProc}`** tag by itself or in combination with other tags to close via the process. Note that if the emulator spawned other processes/windows, these may need to be shutdown independently using the other methods.

- **`{Event}{Esc}`** – Send Esc with default delay of -1 and duration 110 ms (MAME 0.172+).
- **`{Event}!{F4}`** – Send Alt-F4 with default delay (ZSNES).
- **`{Event,-1,-1}!{F4}`** – Send Alt-F4 with a delay delay of -1 and duration -1 (Demul).
- **`!F4`** – Send Alt-F4 as a standard keystroke (Altirra).
- **`!{F4 2}`** – Send Alt-F4 twice (Virtual Pinball).
- **`{WinClose, Visual Pinball Player,Visual Pinball}`** – Send close messages to two comma separated windows (alternate)

	• **^q** – Send Ctrl-q (Stella). • **^{F9}** – Send Ctrl-F9 (DOSBox). • **{Event}{Esc}|{WinClose,Dolphin 5.0}** – Send Esc followed by a command to close window with title: "Dolphin 5.0" (Dolphin)
map_pause=[key or action]	*Optional*. Specifies what action or keystroke/key-sequence occurs when the **pause_key** (defined in **[General]**) is pressed. Like the **map_exit** value, **map_pause** may consist of either one or two strings. If two, then they are separated by a pipe ("|") character. The part to the left of the pipe is always executed first, and the part to the right (if defined) is executed second. If tied to a physical key, the left sequence is executed when the button is pushed down, and the right sequence when the button is released. For example, to toggle the p key, **map_pause** could be set to **{p DownR}|{p up}**. However, since many pause keys are performed by simply toggling a key, a short way to specify the same sequence is **{Toggle}p**. If no pause function is available in an emulator or it cannot be made to work, a synthetic pause, which pauses the windows process of the emulator, may be specified using **{PauseProc}**. *Note, however, that using **{PauseProc}** can have side-effects including the inability to page through **show_on_pause** assets, stuttering when returning to game, crashes to desktop, etc. So using other options is always preferred.* Here are some common settings for **map_pause** (also see examples): • **{Toggle}p** - Toggle the 'p' key (MAME versions 172 and up) • **{Blind}!{Pause}** - Toggle Alt-Pause (DOSBox) • **{Blind}{F9}** - Send an F9 message (Altirra) • **{Toggle}Pause** - Toggle the Pause key (Stella) *When using the following option, exercise caution (see notes above), and test very carefully.* • **{PauseProc}** - Synthetic pause (forcefully pause the emulator process) (Any)
rule_*n*=<[function]> arg1 <[policy]> arg2	*Optional.* arcadeEIP supports both built-in and custom rules. One or more of these rules may be included in the emulator's configuration. Rules are executed in order of their sequence (i.e. **rule_1**, **rule_2**, etc.) which must begin with 1 and have no gaps in the sequence. See the *RulesReference.pdf* document for more information about rules.
clean_lists=[-1, 0, 1, or 2]	*Optional*. Override the **clean_lists** setting for this **[System]**. See the **[Picker_Preferences]** section for more information about this setting.
Any key from the [Key_Map] section	As described in the **[Key_Map]** section, any key-map value may be copied to the **[System]** section. When this is done, the value in the **[System]** section will override the value in the **[Key_Map]** section while the **[System]** section's emulator is running. It is

	recommended that if you need to accommodate different key maps for the picker menu when running under different control panel profiles for various emulators, that you use this method since it does not incur a loading time penalty.
Any key from the [XInput] section	As described in the **[XInput]** section, any key-map value may be copied to the **[System]** section. When this is done, the value in the **[System]** section will override the value in the **[XInput]** section while the **[System]** section's emulator is running. It is recommended that if you need to accommodate different XInput profiles when running under different control panel profiles for various emulators, for example a Fight Stick for arcade games and a Gamepad for console games.
use_hi2txt=[0 or 1]	*Optional*. Note that this setting is for MAME only. If **use_hi2txt** is set to 1, then the 3rd party hi2txt software will be used to show the current game's high score (and player initials) in the bottom scorecard area of the marquee screen. If there is no high score, or if hi2txt cannot resolve a high score, then the marquee image will be displayed normally. Use of this feature requires the hi2txt software to be installed to a subfolder of arcadeEIP called **\hi2txt**. Please see the mame plugin documentation (https://docs.mamedev.org/plugins/hiscore.html), and the hi2txt project documentation (https://greatstoneex.github.io/hi2txt-doc/doc/index.html) for further information. Also see Quick Start 2 – MARQUEE.txt in the root folder of arcadeEIP for some additional setup information. *Note that hi2txt is hit-or-miss on what games it works with. If you encounter problems with certain games, it is far more likely that it is an issue with hi2txt than a problem with arcadeEIP.*

EXAMPLES:

`sys_key=mame`	Specifies "mame" as the unique identifier for this emulator.
`asset_name=MAME`	Sets the string that will be inserted into all **[asset_name]** templates for this emulator to "MAME".
`asset_name=Sega_Naomi_Analog,Sega_Naomi`	Sets the string that will be inserted into all **[asset_name]** templates for this emulator to "Sega_Naomi_Analog", *and* the **[global_asset_name]** to "Sega_Naomi".
`proper_name=MAME`	Sets the "human" name (and value of **[proper_name]** to "MAME"
`exe_full_path=C:\Emulators\MAME\mame64.exe|min`	Sets full path to the MAME executable. The "min" (minimize) token helps hide the console in MAME.
`exe_full_path=D:\Emulators\P64\Project64.exe|nohide`	Sets full path to the Project64 executable. The "nohide" token prevents this particular emulator from "hiding" behind a black screen (at the possible expense of being able to see the desktop while it launches).
`search_path=C:\Stella\roms\*.bin`	Enables searching for partial rom name among all *.bin files in this folder.

`search_path=C:\Stella\roms\*.bin,zip`	Enables searching for partial rom name among all *.bin and *.zip files in this folder.
`search_path=C:\Stella\2kroms\*.bin,zip||C:\Stella\4kroms\*.bin|zip`	Enables searching for the partial rom name among all *.bin and *.zip files in the two specified folders.
`search_path=[root_folder]_*\roms\*.zip`	Enables searching in all **`\rom`** subfolders of the patterned root folder. Used by the **`<change_root>`** rule in support of parallel emulator versions (see Rules Reference Guide).
`param_list=[rom] -autosave -skip_gameinfo`	Typical parameter list for MAME including a **`[rom]`** template filled in by data from the command line.
`param_list=[rom] -skip_gameinfo <-video,1>`	A parameter list that includes a passthrough parameter, in this case, **`-video`** with one argument. This is the parameter that will be used by eip.exe if it is present in the command line. For example, if **`-video ddraw`** is found in the command line produced by the front-end, it will be added to the list of parameters that eip.exe sends to the emulator. If it is not, then nothing will be added.
`mameinfo_folder=.\Assets\MAME\Meta\Info`	Specifies location of **`history.xml`** and **`mameinfo.dat`** files.
`logo_folder=D:\Assets\[asset_name]\Logos`	A path to the game logo png files using an optional **`[asset_name]`** template. Used by bookend screens. Not needed if already defined in **`[Global_Assets]`**.
`system_logo=D:\Assets\[asset_name]\System_Logo\system_logo.png`	A path to the system logo png file using the **`[asset_name]`** template. Used by bookend screens. Not needed if already defined in **`[Global_Assets]`**.
`show_on_pause=logo`	Display a game or system logo on pause Not needed if already defined in **`[Global_Assets]`**.
`run_apps=ultramap,winipac`	Specifies that the ultramap and winipac apps should be run prior to loading the emulator.
`run_apps=ultramap,joytokey,-joytokey`	Specifies that the ultramap and joytokey apps should be run prior to loading the emulator (using the **`[asset_name]`** of the emulator, if used by the application); and specifies that the joytokey app should again be run after the emulator exits, but this time using the **`[asset_name]`** of the front-end (to, for example, set joytokey's profile back to the default).
`run_apps=ultramap,delay(2000),winipac`	Prior to running the emulator, runs the ultramap application, waits for 2 seconds, then runs winipac.
`run_apps=ultramap,send({F3},8000)`	Prior to running the emulator, runs the ultramap application, then triggers an asynchronous wait period of 8 seconds (while emulator is being loaded) after which it will send an {F3} function key-press message.
`run_apps=ultramap,winipac,kill(GameExHideOS)`	Prior to launching the emulator, in addition to running the ultramap and winipac applications, also kill the GameExHideOS window.
`pause_apps=pauseledblinky,-unpauseledbliny`	Runs pauseledblinky application on pause and unpauseledblinky application on unpause.
`rule_1=<remove_param> -autosave <if_rom_in_list> paperboy,eyes,blitz`	Specifies that the **`-autosave`** parameter should be removed for these listed roms. See rules ref.
`map_exit={Event}{Esc}`	Send Esc when the **`exit_key`** is pressed.
`map_exit={EndProc}`	Forcefully terminates the emulator's process when the **`exit_key`** pressed.

```
map_pause={PauseProc}
map_pause={Toggle}p
next_key=v
previous_key=c
invert_page_buttons=1
```

Pause the emulator's process when the **`pause_key`** in the **`[General]`** section is pressed.
Toggle the 'p' key to when the **`pause_key`** is pressed.
Override **`next_key`** value from **`[Key_Map]`** to use the "v" key for this system.
Override **`previous_key`** value from **`[Key_Map]`** to use the "c" key for this system.
Swap the page buttons used by an XInput controller for this system.

[Application_<app_key>]

Use these indexed sections to define applications. Add additional sections by replacing <app_key> in each section name with the app_key text in the section.

app_key=[text]	***Required***. Must contain a *unique* id for this application (unique among all **fe_key**, **app_key**, **list_key**, and **sys_key** values in this configuration). Will match the name given in the section header. Do not use spaces, and only use characters acceptable in a filename.
exe_full_path=[exe_path]<,min><,nowait>	***Required***. Full path to the application's executable. This may be followed by comma (",") separated tags "min" (to run run the application minimized), and/or "nowait" (to make arcadeEIP not wait till the application exits before continuing). The order of these tags does not matter.
proper_name=[text]	*Optional:* Sets the string that will be inserted into the **[proper_name]** template for this application. This is the "human" name of the application and may contain spaces.
profile=<file_specification>	*Optional.* File name with full path recommended. Using a full path enables file validation and "smart" profile finding options. *Please read the following information carefully as the **smart_name** feature is an important feature of arcadeEIP, and can be quite powerful once you realize what it does for you.* *This field supports the following templates*: **[rom_name]** : Replace template with the **rom_name**. **[asset_name]** : Replace template with the **asset_name** for this emulator or the front_end depending on whose **run_apps** setting called the application. **\[smart_name]** : If prefixed with a backslash, will replace template with the **rom_name**. If file is not found, will replace the template with the emulator *or* front-end's **asset_name**. If file is still not found, will replace the template with the string "default" and search *in the parent folder (see examples below).* ***Note that when the [smart_name] template is used, the file search will also search in subfolders of the target folder.*** This allows for profile files to be organized in subfolders according to type if desired. **_[smart_name]** : If prefixed with an underscore, will replace template with the **rom_name**. If file is not found, will replace the template with the emulator *or* front-end's **asset_name** stripped of any preceding text. If file is still not found, will replace the template with the string "default" stripped of any preceding text (*see examples below*).

param_list=[parameters_list]	*Optional.* Parameters for the application If a profile is defined, it should include the **[profile]** template. *This field supports the following templates*: **[profile]** : Replace template with the **profile** defined in this section. **[rom_name]** : Replace template with the **rom_name.** **[rom_file]** : Replace template with **rom_file.** **[rom_full_path]** : Replace template with **rom_full_path.** **[rom_folder]** : Replace template with **rom_folder.** **[rom_folder_parent]** : Replace template with **rom_folder_parent.** **[asset_name]** : Replace template with the **asset_name** for the emulator or the front-end depending on whose **run_apps** setting called the application.

EXAMPLES:

app_key=ultramap	Specifies the key for the ultramap application.
exe_full_path=C:\Ultramap\Ultramap.exe	Path to Ultramap.exe. By default will wait until app terminates before continuing.
proper_name=Ultramap	Set the "human" name of the application to "Ultramap"
exe_full_path=C:\JoyToKey\JoyToKey.exe|nowait	Path to JoyToKey.exe. Since JoyToKey runs resident, we will not wait for the app to terminate before continuing.
profile=D:\Ultrastick\Profiles\[asset_name]\[smart_name].ugc	Given **asset_name=MAME** and **rom_name=berzerk**, will yield: **D:\Ultrastick\Profiles\MAME\berzerk.ugc** If exists. Otherwise… **D:\Ultrastick\Profiles\MAME\MAME.ugc** If exists. Otherwise… **D:\Ultrastick\Profiles\default.ugc** Finally.
profile=D:\Ultrastick\Profiles\[asset_name]\[smart_name].ugc	Given **asset_name=BigBox**, will yield: **D:\Ultrastick\Profiles\BigBox\BigBox.ugc** If exists. Otherwise… **D:\Ultrastick\Profiles\default.ugc** Finally.
profile=D:\Ultrastick\Profiles\[asset_name]_[smart_name].ugc	Given **asset_name=MAME** and **rom_name=berzerk**, will yield: **D:\Utilities\JoyToKey\MAME_berzerk.ugc** If exists. Otherwise… **D:\Utilities\JoyToKey\MAME.ugc** If exists. Otherwise… **D:\Ultrastick\Profiles\default.ugc** Finally.

`profile=D:\Ultrastick\Profiles\[asset_name]_[smart_name].ugc`	Given **`asset_name=BigBox`**, will yield: **`D:\Utilities\JoyToKey\BigBox.ugc`** If exists. Otherwise… **`D:\Ultrastick\Profiles\default.ugc`** Finally.
`param_list=[profile]`	Will run the executable with the specified profile as a parameter.
`param_list=-option1 -option2`	Will run executable with just the specified parameters (if no profile).