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 can always generate a 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 they already exist.

```
>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 a short tutorial that you can use to get a working MAME configuration operating quickly. It also 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 launch front-ends such as GameEx or 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 setup 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 whle 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 and all its parameters are optional.

debug_mode=[fe_key or "off"]	*Optional*: Run in debug mode. *Must* set this to a valid **fe_key** value (use "os" if testing from the command-line). To disable, set to nothing, "off", or remove setting. Note that this can also be specified on the command-line if preferred (see Command-Line Reference Guide.pdf). Debug mode is an important "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. Beep upon startup and termination of **eip.exe**. This can be useful in knowing 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]	This setting activates or inactivates a fade-in effect for the start bookend screen when switching games. It will also activate/inactivate fade effects during marquee transitions as well. The default is 1 (on). When turned off (0) transitions will be instant. Note that this setting can also be used in the [Marquee] section if, for example, you would like the behavior different for the marquee screen vs. the bookend screens.
seed_rom=<vector>	*Optional:* This setting specifies the rom that will be used if arcadeEIP is launched without any other rom specified on the command line or if there are no roms defined in the attract list. The form is **<sys_key> <rom_file>**.
start_seed=<0 or 1>	*Optional:* Default is 0. Set this option to 1 to force arcadeEIP to always use the seed rom upon startup. When set to 0 (the default) the startup rom will be obtained from the current attract list.
sound_device=[integer]	*Optional:* Default is 1. This setting sets the sound device that arcadeEIP will use. Ordinarily, you will want this set to 1, which corresponds to the default audio device. 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 numeric values 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. 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. See **sound.txt** for more information about this as well.*
sync_volume=[0 or 1]	*Optional:* Default is 0. Note that This setting is only useful when **sound_device** is set to a value other than 1. 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]	This setting determines whether license text is displayed. Set **show_license=1** (the default) 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 hiding the license text is only possible with the free hobbyist or a commercial license file. See the license.txt file for more information about obtaining liceneses.*
hotstring_prefix=[string]	This setting determines the unique string that all hotstring, direct-switch filenames must start with. By default this is set to **hotstring_prefix=@!@** 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 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.
`alpha_effect=0`	Turn off fade effect in bookend screens and marquee.
`hotstring_prefix=@!@`	All hotstring direct-switch files must start with the prefix "@!@"

[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**. 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*: 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, preventing it from displaying any startup prompts.
autoexit=[0 or 1]	*Optional:* Default is 0. Setting this value to 1 will cause cause **eip.exe** to automatically close **marquee.exe** when it terminates. If this option is set to 0, **marquee.exe** will display **default.png** (if found) or perform a screen blank if **default.png** is not found, but still remain resident when **eip.exe** is exited.
interpolation=[0-7]	*Optional:* Sets the interpolation used by the GDI+ image scaler. Default is 2; however, other values might be sharper/preferred depending on 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:* 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.

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

[Key_Map]

The **[Key_Map]** section contains all key mappings 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 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 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.
Picker Key – work only in the game selection screen.

panic_key=[key]	*General Key:* Default is Ctrl-Alt-k. This key will immediately close both arcadeEIP and the currently running emulator/game when it can. It will also try to restore the mouse pointer and restore all resoures. 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 Ctrl-Alt-l. This setting toggles a feature that permits the user to pause or unpause the attract timer. This will have the effect of temporarily turning off automatic game switching and locking the arcade machine to the currently displayed game (for this session only). The locked or unlocked states are announced by both voice and by a text overlay that briefly displays the mode. Also see the **pause_lock** in the **[Auto_Switch]** section for an alternate way of invoking this feature.
pause_key=[key]	*Live Key:* Default is ',' (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 thel **[System]** sections is for.
exit_key=[key]	*General Key:* Default is '.' (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 send 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.

menu_toggle_key==[key]<,[key]>…	*Live Key:* Use this key (default is the apostrophe key, ‘) to toggle the game picker overlay menu screen. *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:* Use this key (default, “=”) 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. This key is primarily intended for use with external remote devices that need provision for discrete input.
menu_hide_key==[key]<,[key]>…	*Live Key:* Use this key (default, “-“) to hide the game picker screen. Unlike **menu_toggle**, this is not a toggle but only hides the menu 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]>…	*Live Key:* This key (default, “~”) exits arcadeEIP and the currently running emulator. It’s 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.
magic_key=[key]	*General Key:* The “magic” key can be used to toggle the game selection screen on/off and perform another 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:* Use this key to perform a direct-switch back to the previous game in the Auto-Switch list. 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. By default, either the “q’ or “[“ key may be used (“q” is recommended if you use Stella)
next_key=[key]<,[key]>…	*Live Key:* 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. By default, either the "w" or "]" key may be used ("w" is recommended if you use Stella)
launch_key=[key]<,[key]>…	*Picker Key:* Defalut LCtrl. While in the game selection overlay screen, use this key to launch the 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_mode_key=[key]<,[key]>…	*Picker Key:* Default: LAlt. While in the game selection screen, use the **list_mode_key** to switch between the full game list view and the secondary "picks" view on the currently displayed list. When the secondary list is being displayed, a star will appear next to the list name. Note that by default, this is assigned to the left-alt key, which is the same as the player one, button two key in MAME.
add_remove_key=[key]<,[key]>…	*Picker Key:* Default: Space. While in the game selection screen, use the **add_remove_key** to add/remove games from a system's secondary "picks" list, or from a targeted custom list if you're in edit mode. For example, while in the full list view of a system, press the add/remove key, and the game will be added to the secondary list (it will also be added to "All Picks" if not already there). To remove the game from the secondary list, switch to it using the **list_key**, select a game in that list, then press the add/remove key (it will also be removed from "All Picks" (and possibly "Top Picks") if no other list has designated the game as a pick). Note that by default, this is assigned to the spacebar, which is the same as the player one, button three key in MAME.
edit_mode_key=[key]<,[key]>…	*Picker Key:* Default LShift. 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 All/Top Picks list in the picker menu. 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. Note that by default, this 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]>**… **home_key=[key]<,[key]>**… **end_key=[key]<,[key]>**…	*Picker Keys:* While in the game selection overlay screen, use these keys to navigate. 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 and home/end provide speedier navigation through long lists. By default, page-up and page-down, in addition to being assigned to the typical PgUp and PgDn keys are also assigned to the 1 and 2 keys, which are the

	same as the player 1 and player 2 start buttons 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.
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]>…**	*Picker Keys:* While in the game selection overlay screen, use these keys to copy or remove games that have the same assigned attribute as the selected game. For example, if you are in the master list of a system and the game currently selected is in the *Shooter* genre, then pressing the **genre_key** ("g" by default) will copy all games sharing that genre to the secondary ("picks") list 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" list (or in a custom list while in edit mode), then all *Shooter* games will be removed from that list. This will work similarly for all other 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[u]blisher as selected game — default "u" **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 Launch Box integration (e.g. year, publisher, developer, and players). Genre can be obtained with Launch Box 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]>…**	*Picker Keys:* 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. **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 stars 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)
`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)

[Picker_Preferences]

The **[Picker_Preferences]** section contains settings which control various behaviours and graphical attributes of the Quick-Switch game selection menu.

This section is required and and all its parameters are required.

kiosk_mode=[0 or 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 invoked on a short (i.e. ordinary) press of the **magic_key** and the exit function will be performed on a long press (*note that only exit is supported as the secondary function in mode 1*). 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 press, 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 operating as a launcher under the control of another Front End. This setting (and **magic_delay** above) may be overridden in the [Front_End] sections.
magic_delay=[milliseconds] *Use 0 or >500 for best results*	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]	When set to 1 (the default) each list in the game selection screen will display a star in front of its name when toggled to display "picks". If **mark_picks** is set to 0, however, then these lists will display a star in front of their names when showing their master list instead. This setting is intended to declutter the view in cases where lists tend to be kept in their "picks" state.
joy_vert=["all" or "off"] **joy_horiz=["all" or "off"]** **mouse_vert=["all" or "off"]**	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.

mouse_horiz=["all" or "off"]	***Troubleshooting note:*** *If you ever encounter an issue with menus drifiting or acting "by themselves", try setting these values to "off" to see if it may be a controller calibration issue.*
default_list=<"picks"> or <list_key> or <sys_key>	This setting determines the list that the game selection screen will display by default. Use "**picks**" for the All/Top PIcks list (the default), or chose the **list_key** of a **[List]** or the **sys_key** of a **[System]**.
all_picks_label=[text] **top_picks_label=[text]**	These settings determine the text that will be displayed for the all picks and top picks lists. By default **all_picks_label** is set to "All Picks" and **top_picks_label** is set to "Top Picks.
hide_systems=<sys_key><,sys_key>…	This setting will hide one or more systems from the game selection screen. Note that only **sys_key** values from **[System]** sections are allowed (and not, for example, **list_key** values from **[Lists]**). 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 All/Top 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]	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 parenthesis 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 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 parenthesis (( )) 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]**	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]**	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 side-effects.
asset_name=[text]	Contol panel profiles can be defined for the Quick-Switch game picker screen just as in other systems. 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 you may also assign that **asset_name** here. The the default **asset_name** is "**EIP_Picker**", and it is advised to keep this name unless you strongly desire another. ***Be aware**, however, that while changing your control panel profiles for the picker is supported, it is not really recommended unless you absolutely need it, since profiles can take a few seconds to load, and thus will affect how quickly the picker can be displayed. To avoid this, it is always preferrable to use key map overrides in the appropriate **[System]** sections(as described above in the **[Key_Map]** section) instead of profile switching to avoid the performance impact.*
run_apps=<app_key><,app_key>…	Defines one or more apps (defined in the **[Application]** sections that will be automatically run when switching to the picker menu. Normally, only required 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).

EXAMPLES:

`magic_delay=0`	Display the Quick-Switch game picker menu immediately
`magic_delay=500`	Display picker (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 picker menu.
`joy_horiz=off`	Joystick control for horizontal navigation in the picker menu is disabled.
`mouse_vert=all`	All detected mouse controllers are enabled for vertical navigation in the picker.
`mouse_horiz=all`	All detected mouse controllers are enabled for horizontal navigation in the picker.
`default_list=picks`	Make All/Top Picks the default list in the picker menu.
`default_list=mame`	Make the MAME system the default list in the picker menu.
`default_list=autoswitch`	Make the Auto-Switch list the default list in the picker menu.
`hide_systems=vic20,c64`	Hide the vic20 and c64 systems (and their games) from the picker menu.

`top_rows=12`	Display 12 rows of games above the Game Name bar in the picker menu.
`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 sized 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`	Set the width of the Game Name relative to the images on either side.
`asset_name=EIP_Picker`	Set the name which will fill the [**`asset_name`**] template as "EIP_Picker"
`run_apps=ultramap,winipac`	Set the applications that will run when switching to the picker menu (see notes).

[Auto_Switch]

The **[Auto_Switch]** 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]	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]** 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]	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]	If **mute** is set to 0, then all system sound will be disabled while Auto-Switch mode is in effect. If it is set to 1, then sound always remains turned on. ***Note*** *that if this is set to zero 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 intial sound effects might be clipped).*
delay=[seconds]	Use the **delay=** setting to turn the auto-switch mode feature on or off. Set delay=0 to turn off, or delay > 20 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. This means that 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 intervening activity from constantly pusing 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=[setting]	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 a 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 swtich 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]	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]	Default 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.

logo_folder=[path]	*Advised but optional.* 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:* 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.* Currently only needed if you use dynamic marquees (see **[Marquee]** section). Specifies the folder in which marquee assets may be found. 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:* 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.* 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.
meta_names=[full file path]	*Advised but optional.* 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) For example: **D:\Assets\[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 definition: **fe_key=os**.
asset_name=[text]	*Optional*. 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:* 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*. Displays the startup bookend (loading) screen before the emulator runs for a minimum of **[sec]** seconds. Default is -1 (off). 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.* Displays the exit bookend screen after the emulator exits. Takes same arguments as **start_screen**
hide_cursor=[0 or 1]	*Optional*. Hides (1) or shows (0) cursor 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.
hide_taskbar=[0 or 1]	*Optional*. 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. Technically, this should only be needed when running emulators that use the "nohide" option tag.
run_apps=<app_key><,app_key>…	*Optional.* Defines one or more apps (defined in the **[Application_*n*]** sections) that will be run whenever returning to the os. In addition, the following prefixes are supported. "+" : Applications prefixed with a "+" will be run before any apps called in an emulator section.
show_on_pause=[asset type]	*Optional:* Specifies what kind of asset(s) will be displayed when the pause key is pressed. By default this will be set to **show_on_pause=controls,info**. Assets are displayed in the order indicated. Use the left/right keys or joysticks to move from page-to-page while paused. The default setting means that images specified by the controls_folder= and/or **system_controls=** settings (i.e. control panel map files), will be displayed first, followed by any available info for

	the game (see below). You may optionally add or specify other existing types such as **logo** or **marquee**. If an asset type is not available, that is ok. It will simply not be displayed. 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 these two keys… **3dbox_folder=**… **system_3dbox=**… …you would be able to specify, **show_on_pause=3dbox** to display 3D Boxes 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. **info** The "info" option is special reserved type (not an image type), and is designed to show 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. 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, the .dat format is not supported. For more information about this setting, see **mameinfo_folder=** entry in the **[Systems]** section below.
magic_delay=[milliseconds] *Use 0 or >500 for best results*	This setting (and **kiosk_mode** below) may be be used to override the setting in the [Picker_Preference] sections.
kiosk_mode=[0 or 1]	This setting (and **magic_delay** above) may be 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 map image 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 name given in 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.
`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 foruse in the default **`[Front_End]`** section may be used in these 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.

[List_<*list_key*>] Each **[List_<sys_key>]** section defines an 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 name given in 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=bu`	Prefixes the proper name with "bu" prior to sorting the lists for display 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 name given in section header. Do not use spaces, and only use characters acceptable in a filename.
asset_name=[text]<,text>	*Optional:* 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 frequest 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:* 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.

sort_prefix=[text]	*Optional:* Default is blank. 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=b** (or any text that would precede "Daphne" alphabetically) to the section defining the *MAME* list.
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—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. Options 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 three that exist currently. Cannot be manually combined. Note that **window_name** (see next setting) must be correctly set in the section in order for A-C to work. <u>Variants:</u> **winset_A**: Attempts to hide window title, menu, and borders. **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 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 loose 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 command. 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 **eip.exe [sys_key] -list** Keep in mind that partial matching can be inexact.
mameinfo_folder=[path]	*Optional.* 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 **logo_folder** setting description above in the [**Global_Assets]** section for more information.
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 the **[Global_Assets]** section. See the **show_on_pause** setting description above in the [**Global_Assets**] section for more information.
param_list=[parameter_list]	*Optional.* Specifies the 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 Passthrough Parameters 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 a 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 both **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 modifers **{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 combintation 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 **`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}**. Here are some common settings for **map_esc** (also see examples): • **{PauseProc}** - Synthetic pause (forcefully pause the emulator process) (Any) • **{Blind}!{Pause}** - Toggle Alt-Pause (DOSBox) • **{Toggle}p** - Toggle the 'p' key (MAME) • **{Blind}{F9}** - Send an F9 message (Altirra) • **{Toggle}Pause** - Toggle the Pause key (Stella)
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 you 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 an loading time penalties.
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) at the bottom 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.

	In some cases, hi2txt processing can cause the corruption of a game's nvram files, especially if there is a version mismatch between the current version of MAME and the one the hi file was written to work with. If you are having this issue, you can omit selected roms from being processed by listing them in a file called **`omit.txt`**, and placing this file in the **`\hi2txt`** folder. When specifying roms in this file, each rom should be on a separate line, and only the root name of the rom should be used. For example, **`berserk`** **`dkong`** If arcadeEIP sees the rom name in this file, it will skip any hi2txt processing. *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 execuatable. 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 parameter that will be used as a parameter 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 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\syste
m_logo.png
```
A path to 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}
```
Pause the emulator's process when the **pause_key** in the **[General]** section is pressed.

```
map_pause={Toggle}p
```
Toggle the 'p' key to when the **pause_key** is pressed.

```
next_key=v
```
Override **next_key** value from **[Key_Map]** to use the "v" key for this system.

```
previous_key=c
```
Override **previous_key** value from **[Key_Map]** to use the "c" key 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 name given in section header. Do not use spaces, and only use characters acceptable in a filename.
exe_full_path=[exe_path]<|min><|nowait> ***the delimiter used is determined by the delimiter= setting in [General].*	***Required***. Full path to the application's executable. This may be followed by pipe ("|") 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 till 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 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).