

IV/PLAY USER GUIDE

John L. Hardy IV

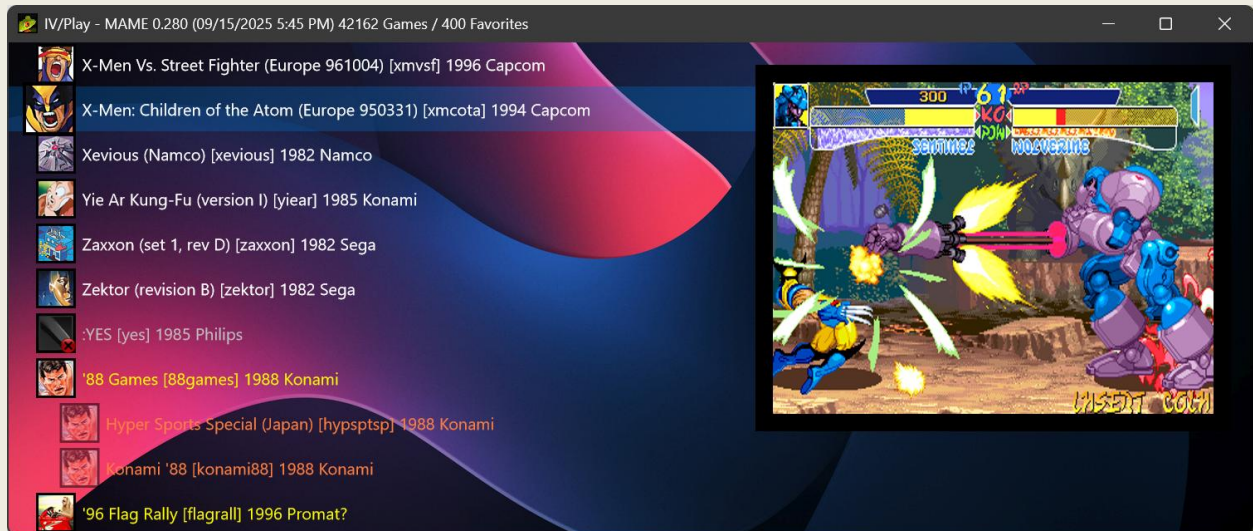


TABLE OF CONTENTS

Overview.....	4
IV/Play vs. MAMEUI.....	4
Requirements	4
Installation	4
Resources.....	5
Major Application Functions.....	5
Import XML Data from MAME.exe.....	5
Configuration Screen (F1).....	5
Display XML Data in UI	5
Title Bar	5
Game List	5
Favorites	6
Icons.....	6
Art Area.....	7
Text Filter Dialogue	7
Background Selection and Randomization	7
Properties View.....	8
Launch Game.....	8
Accelerator / Navigation Keys	8
External Files.....	9
IV-Play.cfg.....	10
IV-Play.dB	10
Updates	10
IV/Play 1.5 Features	10
Unlimited Art Types	10
Favorites Toggle	11

Non-Working Game Art/History Support	11
Command Line Override	11
IV/Play 1.6 Features	11
IV/Play 1.7 Features	11
Clone and Non-Working Filters.....	11
IV/Play 1.8 Features	11
IV/Play 2.0 Features	11
Move from .NET 4.5 to .NET 8.0 & Exit from WPFWrapper to Pure WinForm	12
GPU Render Pipeline	12
LiteDB to SQLite	12
Support for High DPI Monitors / Large Icon Mode / Scaling Improvements	12
Support for Software List Devices.....	13
Custom Game List Support	13
Grid View	13
Background Brightness Detection Mode	13
Image Atlas for Icons and Binary Versions of History.XML and MAMEINFO.DAT	13
Additional Features.....	14
IV/Play 2.1.0 Features	14
Move from .NET 8 to .NET 9	14
Additional Features.....	14
Notes/Hints.....	15
Version History & Development Timeline	15
Appendix.....	15
Architecture	15
Filter and Search Guide	17
Credits.....	20

OVERVIEW

IV/Play (pronounced 'Four Play') is a desktop/keyboard-oriented GUI front-end for [MAME™](#). It was designed, and its coding commissioned, in 2011 by original MAMEUI team member and non-programmer, John IV, from his original concepts of 2006. It has a particular feature-set, is keyboard driven, and utilises many of the navigation and keyboard short cuts of MAMEUI. It is targeted towards the latest version of Microsoft Windows 11 (2025). IV/Play is decoupled from setting MAME options directly, using its (MAME's) MAME.ini in an effort to future proof and guard against core command line changes and updates. It is self-contained and portable for easy transport and mobility.

In 2025, with the rise of AI LLMs, John IV utilised a group of them in a pseudo 'pair-programming' mode to undertake the modernization of the 14-year-old codebase from its .NET 4.5 WPFWrapper origins to its current state, rebuilt entirely using WinForms, with WPF fully deprecated. It provides a high performance, RAM based experience on i7-12700K, 64GB RAM, GPU level machines. The goal is near instant launch, game list population, and zero slowdown displaying any art assets (icons, snaps, flyers, bkgounds, etc.) utilizing as much RAM and graphic (GPU) hardware as needed; employing modern design techniques and technologies.

Naming conventions: The application name is 'IV/Play' but due to WinOS file-naming limitations, 'IV-Play' is used where necessary.

IV/PLAY VS. MAMEUI

MAMEUI is an integrated UI to [MAME](#) and is compiled using the base source. It was the first 'official' MAME port to the Windows platform and the most popular version of MAME for decades. It leveraged a lot of the underlying MAME code and depended on it to produce its UI and functionality. This was its strength and weakness. As MAMEUI matured (b. 1997), the original development team waned and it often took a long time to catch up with the sweeping core changes that periodically took place in the underlying MAME engine. While the number of supported arcade games, gambling 'Fruit Machines', computers, calculators, consoles, and miscellaneous other ROM based electronics neared 40,000 sets, the UI slowed down when all the icons and screenshots were processed by the near 3-decade old controls. This introduced memory leaks over time and the user experience degraded. Changes to the core at the time also resulted in huge memory footprints and 5x longer load times.

IV/Play was born out of the desire to continue a similar 'classic' desktop interface to MAMEUI without the attendant issues of being coupled to the core.

REQUIREMENTS

Windows 11 is the preferred platform. Faster modern processors and solid-state drives (SSDs / NVMe) help considerably with image/resource loading speed and are strongly recommended. IV/Play 2.0.0 is fully DPI aware and will scale its UI and elements for a 4K monitor for example. IV/Play is built using .NET 9 and will prompt to download it from Microsoft if not present. IV/Play, as mentioned in the overview is now (2025) being optimized for high end systems, which include the latest processors, copious amounts of RAM, and GPUs to drive its performant mission.

INSTALLATION

The most straightforward installation option is to place the IV/Play files into the top level of an existing MAME directory structure alongside the respective .exe. IV/Play will utilise the structure's icons, art types, and bkground folders of an old MAMEUI installation. Baseline MAME binaries can be used or versions of MAMEUI if desired; IV/Play can link to either .exe. IV/Play does not rely on registry settings or an install so it is easily portable on external media like a thumb drive with MAME and attendant files. IV/Play will create its own named directory under its location for its various assets, but also maintain a same level favorites.ini file, custom game list and its *.cfg file, which allow them to survive the asset directory removal during testing or refresh. The attendant e_sqlite3.dll must also be present with the .exe.

RESOURCES

The current build, icons, and snapshots for IV/Play are found at the [IV/Play homepage](#).

MAJOR APPLICATION FUNCTIONS

IV/Play is, at its core, a game launcher. It depends on the presence of a MAME.exe to import its XML (populating its list view and properties), and to pass back its game launch command. IV/Play is designed to be as low maintenance as possible and will not directly set MAME options; edit the baseline MAME.ini for that (though it is possible to use the **F1** command-line override to supersede the MAME.ini).

IMPORT XML DATA FROM MAME.EXE

IV/Play resides in the same directory as the MAME executable or in its own directory. Upon initial launch, it searches the local directory for a MAME version. Failing to find one a dialogue box appears asking to locate the MAME.exe to 'link'. It then grabs the output of the (currently) 290MB MAME XML file (MAME.exe -listxml). **F5** refreshes the XML using the same named MAME.exe, (when a new version of MAME is released for example) while **F4** re-opens the dialogue box to select a different MAMExx.exe. **Note:** IV/Play can utilise any modern version of MAME that produces the XML output needed, including MAMEUI64. The various MAME *.exes are used interchangeably in this document.

CONFIGURATION SCREEN (F1)

F1 brings up the configuration screen. The UI settings are stored in the user editable IV-Play.cfg file, and may be viewed in-session with the **F2** key.

DISPLAY XML DATA IN UI

TITLE BAR

The title bar displays the version of MAME linked along with the number of games supported in that version and favorites in use. The number of favorites displayed will change if the favorites and filter are used.

GAME LIST

The game list displays all the games from the MAME XML output. Clones are automatically offset from their parents (but can have the indent removed in the **F1** screen). The font is selectable as is the color of the parents,

clones, non-working, and favorites. Navigation through the list is via regular direction keys like **up**, **down**, **page up**, **page down**, **home** and **end**.

Display of the year, manufacturer, and machine (internal MAME) name is configurable through **F1**.

Free typing 'donkey kong' in the game list goes to the first instance depending where focus is when typing; if there are no instances downward it will go to the first instance of 'donkey kong' which may wrap around to the top in the favorites.

The Game List can be toggled with **Alt-P** and **TAB** to step through the Game List modes: regular icon list, large icon list, and large icon grid.

FAVORITES

IV/Play supports the use of a `.\favorites.ini` file. This is a text file with one game per line, for example:

```
10yard
1941
dkong
frogger
```

Favorites are toggled by **Alt-D** and can be set in the **F1** configuration screen. The favorites are displayed above the MAME generated game list in alphabetical order. The icons are intentionally **not** offset for favorites to avoid confusion if a clone is added as a favorite and its parent is not there. The favorites can be displayed in different colors than the regular game list (set via **F1** in the configuration dialogue). The **Alt-D** toggle will cycle through favorites off, favorites + game list, favorites by themselves. Non-working and clone games added to favorites survive filters set on the lower game list.

Favorites are added by **Ctrl-D** from the game list and removed by **Ctrl-D** from the favorites area.

Up/Down/Left/Right/Page Up/Page Down behave the same way they do in the game list. A single press of **down arrow** on the last favorite goes to the first regular game list item. **Home** and **End** buttons on the keyboard provide a two-stop navigation system to move between the favorites and main game lists efficiently.

- **End Key:** When the focus is in the favorites list, the first press of **End** jumps to the last favorite. A second press jumps to the very end of the main game list. If the focus is already in the main list, it jumps directly to the end.
- **Home Key:** When the focus is in the main game list, the first press of **Home** jumps to the top of the main list (directly below favorites). A second press jumps to the top of the favorites list. If the focus is already in the favorites, it jumps directly to the top.

ICONS

Each game supported by MAME can have an associated, freestanding *.ico in the `.\icons` directory.

Base icon size and the icon scale presets can be chosen with the drop box in **F1**.

If 'Draw non-working overlay' is toggled on in the configuration menu, a small red circle 'x' will be overlaid on any non-working game in the lower right-hand corner of its icon.

ART AREA

The upper right of the game list is the art area. The art area displays the current art type for the selected game. These are freestanding *.png files that reside in their respective directories and take the name of each game, e.g. dkong.png.

Alt-1 through Alt-0 toggles the various art types (which are added through the **F1** configuration dialogue).

Opacity for the art types is set in **F1** configuration dialogue.

The border width and color for the art can be set with **F1**.

Left clicking on the art area with the cursor toggles the art type.

If a game is displayed for which there is no parent snapshot (for example after a new version of MAME is released and linked, prior to a snapshot pack being created), then nothing displays in the art area, the bkground.png will show through.

IV/Play supports History.XML and MAMEInfo.DAT support files available at their respective sites. They are searched for in the MAME.exe directory on initial IV/Play launch and can be added at any time through the **F1** configuration menu. The History.XML and MAMEInfo.DAT files may also be added manually with the Add DAT file button in **F1**. These are accessible with the 'Peek' option using the tilde key (~) above the tab on most keyboards. Left and right in DAT peek mode sizes the peek overlay horizontally, the **TAB** key cycles through the DAT types while the peek overlay is open.

Snapshots and other art assets are placed in the upper right of the window offset from the edge, and are scaled to the height of the window. If art is wider than 40% of the horizontal width of the window, it will be scaled to fit. Border width, color, and its opacity are set in the **F1** screen.

TEXT FILTER DIALOGUE

The Text Filter dialogue box is accessible with either **Ctrl-F** or **Ctrl-E**. When typing in this field and hitting enter it will search and display resulting hits, like a filter. For example, typing 'donk' would bring up Donkey Kong and all other items with 'donk' in its description including something like 'Crazy Donkey'.

The filter can also take manufacturer, year, driver source for the game, i.e. 'dkong.cpp', and other key elements. Filtered game list is maintained until the filter is cleared. The filter includes favorites if toggled on. The dialogue can be dismissed with **ESC**. Filtered result counts are displayed on the title bar. A filtered view may be backed out of with **ESC**. The filter can also take powerful advanced searches and gives some examples in its drop down. It is able to form these advanced searches based on elements imported from the MAME XML. See the appendix for additional filter syntax and information on natural language 'aliases' for some common search topics.

BACKGROUND SELECTION AND RANDOMIZATION

The lowest art layer of the main list area takes a user defined background *.png image from the .\bkground\ directory. The **F1** configuration dialogue toggles the background image rotation, so IV/Play will use a random *.png from the .\bkground directory on each launch. **Note:** If the bkground image is 'larger' than current game window it will scale to fit the inner window. If the bkground image is smaller than current game window and less than 903px wide and 800 tall, it will tile from upper left. These bounds can be altered in the *.cfg.

PROPERTIES VIEW

Alt-ENTER displays the properties dialogue for the selected game. It can be dismissed with the **ESC** key. The text in each field is highlight-able and can be copied out of if needed.

LAUNCH GAME

Launching a game via Enter or double-clicking sends the simple argument 'MAME.exe dkong' to the linked MAME.exe. The configuration for MAME is done using its own *.ini files. Holding down the **CTRL** key and hitting enter will launch the game and run its last recorded *.inp file if found. By default, a regular launch will record an *.inp of the game session. Holding down **SHIFT** and hitting enter on a software list machine will launch that machine as if it had no cartridge inserted, like an Atari 2600 or Intellivision.

ACCELERATOR / NAVIGATION KEYS

The following are the keyboard shortcuts in IV/Play:

~ (Tilde)	Toggles the DAT peek feature. Use with left and right arrows to size from 40 to 75 and 100 percent UI width. Up, down, Page Up/Down, Home, and End can be used to scroll, as is common in the other text areas of the app. The TAB key will move to the next DAT file and then loop.
Alt-1 through Alt-0	Go directly to art type. (left click on art area) Note: 'None' is always 0.
Alt-D	Toggle the display of favorites.ini. (off, favorites + games, favorites only)
Alt-Enter	Display properties for the game selected.
Alt-I	Cycle through the icon display sizing presets.
Alt-Left Arrow	Exits a DAT overlay, software list view, or clears and active filter.
Alt-P / TAB	Toggle through the different game list views. (right click on UI)
Arrow Left / Right	Navigate one letter's worth of games, from A to B to C, etc. through favorites and down through the main game list eventually looping to the top again.
Arrow Up / Down	Navigate one game at a time.
Ctrl-D	Add or remove a favorite from favorites.ini.
Ctrl-Enter	Launch the inp of a game that has been previously recorded. (note that by default any game launched normally will have its session recorded as an .inp)

Ctrl-F / Ctrl-E	Open text filter dialogue.
Ctrl-R	Select a random game.
Ctrl-Shift-C	Copies the contents of the filter results list to the clip board in tab separated, Excel friendly format.
F1	Display the configuration dialogue.
F2	Launches the IV-Play.log overlay.
F3	Launches the IV-Play.cfg overlay.
F4	Select the MAME.exe.
F5	Refresh the XML import from MAME.exe and rebuild icon / art caches.
F7	Toggle the onscreen diagnostic dashboard for FPS, memory usage for CPU and GPU, and garbage collection.
F8	Factory reset, removes the \IV-Play directory and *.cfg but leaves favorites.ini and the IV-Play.Custom.ini. It will prompt for the action.
Home / End	Move to the beginning / end of the game list, or as a stop at the beginning and end of the favorites (press home/end a second time to navigate out of favorites).
Page Up / Page Down	Navigate one window height worth of games.
Shift-Enter	Launch a softlist machine without entering its softlist (effectively like powering on a console with no cartridge in it).

EXTERNAL FILES

IV/Play.exe can reside in its own folder or along with MAME.exe. It also now creates a sub-folder for its asset files for easy rebuilds. It's recommended to keep the *.inis at the same level as the .exe so they survive directory purges.

C:\games\IV-Play\IV-Play.exe
 C:\games\IV-Play\IV-Play User Guide.pdf
 C:\games\IV-Play\e_sqlite3.dll
 C:\games\IV-Play\Favorites.ini
 C:\games\IV-Play\IV-Play.Custom.ini
 C:\games\IV-Play\IV-Play.cfg
 C:\games\IV-Play\ IV-Play\IV-Play.db
 C:\games\IV-Play\ IV-Play\IV-Play.log.txt
 C:\games\IV-Play\ IV-Play\IV-Play.history.cache
 C:\games\IV-Play\ IV-Play\IV-Play.mameinfo.cache
 C:\games\IV-Play\ IV-Play\IV-Play.icon.atlas.png
 C:\games\IV-Play\ IV-Play\IV-Play.icon.map.json
 C:\games\IV-Play\ IV-Play\IV-Play.icon_atlas.hash

Once linked to a MAME.exe, IV/Play will use its structure for its art, for example:

```
C:\games\MAME\MAME.exe
C:\games\MAME\History.XML
C:\games\MAME\MAMEInfo.DAT
C:\games\MAME\INI\MAME.ini
C:\games\MAME\icons\gamename.ico
C:\games\MAME\snap\gamename.png
C:\games\MAME\flyers\gamename.png
C:\games\MAME\cabinets\gamename.png
C:\games\MAME\PCBs\gamename.png
C:\games\MAME\marquees\gamename.png
C:\games\MAME\titles\gamename.png
C:\games\MAME\cabinets\gamename.png
C:\games\MAME\bkground\bkground.png
```

IV-PLAY.CFG

This file is created on first launch of IV/Play if it is not present. It contains the configuration settings for the application. Delete this file (and the IV-Play sub-directory) to start fresh. Note that the *.cfg file can point to a custom asset directory (asset_directory), which is helpful if a RAM drive or similar technology is being employed for storage.

IV-PLAY.DB

The DB file is an SQLite database created from the contents of the XML output produced by the MAME engine. It is created on first run if not present after asking for the .exe. If IV/Play is run without the *.db in its directory it will ask for a location of the MAME.exe (if not finding it in the same directory) and use that path in the IV-Play.cfg file. **F5** refreshes the XML and updates the *.dat. **F4** re-links to a new MAME.exe. Delete this file to affect a re-build of the database. **Note:** the **e_sqlite3.dll** **must** be in the same directory as the IV/Play .exe for the app to work.

UPDATES

IV/PLAY 1.5 FEATURES

UNLIMITED ART TYPES

IV/Play allows for the display of 'unlimited' art types. In the **F1** configuration dialogue, any directory can be added that contains art types following the *gamename.png* format. IV/Play will automatically add snap, flyers, cabinets, PCB, marquees, cpanel, and titles on initial launch if it finds them. The view order can be set by moving the folders up or down the list. The art types will be assigned **Alt-<X>** shortcut keys depending on position, **Alt-1** through **Alt-0**.

History.XML / MAMEInfo.DAT Support

IV/Play supports History.XML and MAMEInfo.DAT. They are searched for in the MAME.exe directory on initial IV/Play launch and can be added at any time through the **F1** configuration menu. Activation of the view is via the tilde ~ key which opens the DAT 'peek' view. The peek overlay is expanded left or right to multiple sizes and navigation is via the regular arrow keys, **page up** and **down**, and **ctrl-end** and **ctrl-home**. A separate font with size and color is selectable for the text area on the **F1** configuration menu.

FAVORITES TOGGLE

Favorites can be cycled with **Alt-D** through off, on with game-list, and favorites only.

NON-WORKING GAME ART/HISTORY SUPPORT

IV/Play will display icons and any art type for non-working games if they are present. This also allows viewing of History.XML and MAMEInfo.DAT entries for non-working games.

COMMAND LINE OVERRIDE

The command line override in the **F1** configuration dialogue allows the use of various switches to be added to launched games, e.g. -window to play games windowed without having to drop to the mame.ini for editing. Note that command line arguments take precedence over MAME's *.ini processing hierarchy.

IV/PLAY 1.6 FEATURES

Updated to work with new XML output of MAME .162.

IV/PLAY 1.7 FEATURES

CLONE AND NON-WORKING FILTERS

IV/Play now has the ability to filter out clones and/or non-working machines in the **F1** configuration screen.

IV/PLAY 1.8 FEATURES

- New background and modern Windows Segoe UI font as default on first run.
- ESC will exit IV/Play.
- It is now possible to un-indent the clones in **F1**.
- Relative paths are now used in the IV-Play.cfg file, this allows for easy portability/transport via thumb drive or removable storage.
- With 1.8.4.0 added support for the new format of the History file, now an *.XML at www.arcade-history.com.
- With 1.8.5.0 added the ability to change the color of the non-working games/devices in **F1**.

IV/PLAY 2.0 FEATURES

IV/Play sees a significant modernization effort, rebuild, and a full version bump in the summer of 2025.

MOVE FROM .NET 4.5 TO .NET 8.0 & EXIT FROM WPFWRAPPER TO PURE WINFORM

A migration was undertaken successfully to get IV/Play from .NET 4.5 all the way to 8.0 and also to extricate it from the WPFWrapper which had ultimately caused bothersome flickering of the handoff on launch.

GPU RENDER PIPELINE

IV/Play has been converted from a GDI app to utilising a full GPU accelerated pipeline. This leads to considerably smoother scrolling and navigation in-app with art assets appearing instantly from its various caches. Art, icons and backgrounds scale via DirectX High Quality Bicubic instead of straight bilinear filtering, resulting in sharper images on higher resolution monitors.

This architectural change was a strategic decision to resolve a persistent and severe performance bug known as "scrolling judder". The previous GDI+ rendering pipeline was CPU-bound, meaning all drawing operations were handled by the main processor. During fast navigation, this would overwhelm the UI thread, causing the application to stutter or freeze.

The new pipeline offloads all rendering to the system's graphics card (GPU) using DirectX 11 / Direct2D. This frees the CPU from heavy drawing tasks, resulting in a perfectly smooth, zero-latency user experience, even when scrolling rapidly through thousands of games with high-resolution artwork. This one-time processing of assets on startup is an intentional trade-off that enables the application's highly responsive and fluid feel during use.

LITEDB TO SQLITE

To ensure long-term stability and performance, IV/Play's data layer was migrated from the LiteDB engine to SQLite using the powerful Entity Framework Core.

This was a critical architectural change driven by a series of persistent and difficult-to-diagnose crashes originating from the LiteDB query engine. The migration to a mature, industry-standard backend like SQLite completely resolved these stability issues.

SUPPORT FOR HIGH DPI MONITORS / LARGE ICON MODE / SCALING IMPROVEMENTS

- IV/Play is now high DPI aware; it won't be necessary to use Windows 11 compatibility hacks to enable a non-miniscule screen. The UI is now at 1523x648 which is proportionally what it was at 1015x432 on 1080p monitors. Running on 1440DPI and 4K now looks very nice; the fonts are crisp and each art asset looks the way it should instead of being simply stretched after a Windows snapshot. Everything scales properly based on the DPI setting including row height.
- Large icon view is now supported, it can be toggled in the **F1** dialogue and via **Alt-P** or **TAB** as one of the views in the cycle.
- Backgrounds will now scale themselves to the inner window size, a set size floor for that functionality can be altered in the *.cfg so anything smaller than the setting will be tiled. This works particularly well for aspect ratio friendly bkgounds that may be slightly bigger or smaller in their dimensions than the current inner window.
- Long time bug fixed for maintaining a maximized state on re-launch.

- IV/Play no longer locks its assets with file handles, so the snaps viewed in that session can be replaced on disk underneath the app.

SUPPORT FOR SOFTWARE LIST DEVICES

IV/Play can now click/enter on a console like Atari 2600 and it will read its softlist and display as the new game list. The navigation is the same as the game list, though there are no favorites displayed. Hit **Backspace** or **ESC** or **Alt-Left-Arrow** to return to the level above with highlight on the system entered.

CUSTOM GAME LIST SUPPORT

Similar to the favorites.ini, IV/Play now allows the use of the IV-Play.Custom.ini file, selectable from the **F1** dialogue. If chosen, this file takes over the game list and displays entries contained within. This is a single file, but it can be nested based on `[]` or `<>` heading delimiters that will produce the drop-down choice.

```
#Custom GameLists
```

```
<Golden Age>
```

```
amidar
```

```
armora
```

```
astdelux
```

```
asteroid
```

```
<Fighters>
```

```
sf2
```

```
sfa3
```

GRID VIEW

Added the grid view. This will display the games in a grid with machine names under the icons for readability. The grid view behaves in the same fashion as the Game list view. **Alt-P** and **TAB** will cycle through the game list modes: regular icon list, large icon list, and large icon grid.

BACKGROUND BRIGHTNESS DETECTION MODE

IV-Play can now automatically determine if the bkground image is of a very light/bright variety and adjust its fonts to black accordingly (which can still be overridden in the **F1** config).

IMAGE ATLAS FOR ICONS AND BINARY VERSIONS OF HISTORY.XML AND MAMEINFO.DAT

To maximize performance, IV/Play combines all individual game icons into a single large image file called an "icon atlas". This atlas is loaded into GPU memory on startup, allowing for extremely fast icon display and smooth scrolling by eliminating the need to load thousands of individual files from the disk.

To accelerate startup, IV/Play creates optimized cache files for History.XML and MAMEInfo.DAT the first time they are used. On all subsequent launches, the application reads from these pre-parsed files, dramatically reducing load times by avoiding the slow process of reading the large, original text files.

ADDITIONAL FEATURES

- IV/Play.cfg is now alphabetized and sectionalized.
- Via the on-by-default Record-Player toggle, machines launched will have their session *.inp recorded. To play it back, launch the same game a second time with the **CTRL** key held down. Controlled in the *.cfg file (record_player), not directly exposed to **F1**.
- Nearest neighbor scaling is an option now in **F1** (unsmoothed snapshots); it allows a more pixelated scaling algorithm that lends a retro 'air' to the snaps.
- A simulated scanline effect can now be applied to snaps, both over the pixelated and the smooth versions.
- Clone icons and the focus/highlight are now transparent and controlled via settings in the *.cfg file.
- Launching fresh on a machine with a 1080p monitor will alter the DPI settings to the historical dimensions of 1015x432 and alter the icon and row size appropriately. Delete the *.cfg file or Reset to Defaults in **F1** if IV-Play has travelled portably from a 4K system to reset to the 96dpi destination.

IV/PLAY 2.1.0 FEATURES

MOVE FROM .NET 8 TO .NET 9

Upgraded the project from .NET 8 to .NET 9 to benefit from the latest runtime performance enhancements, contributing to a more responsive UI and faster icon cache processing.

ADDITIONAL FEATURES

- Tapping the **Tilde (~)** key will open the DAT 'peek' view where the DAT files are accessible in a temp window. Left and right size the window to 40%, 75%, and full width of the UI. **TAB** cycles through the DATs.
- Copy out of Filter, create a filter search and then **Ctrl-Shift-C** to copy the results to the clipboard in a tabbed, Excel friendly output.
- **F2** (log) and **F3** (config) now activate their 'peek' overlays with navigation the same as the DAT peek.
- **F8** will do a 'factory reset' and remove the contents of the \IV-Play directory including the *.db and art caches. It also removes the *.cfg file, but leaves the favorites.ini and custom game list ini.
- Aspect ratio constrained sizing: If **CTRL** is held down during a sizing event the UI will scale up or down in 2.35:1 TohoScope.
- **F7** displays the performance dashboard/overlay in the upper right corner giving information on memory usage for the system, GPU, FPS, and garbage collection data.
- Distributable is now packaged with AOT (Ahead-of-Time) compilation during publication (vs. JIT) which speeds the initial bring-up of the app.
- **F10** runs a short BVT (Build Verification Test) used in development to exercise basic functionality for breakage.
- 'Include Parents in Filter Results', is now a toggle in **F1**. On by default, if this is set to off, resulting hits searching for variants with clones in the filter will no longer display their parents with them for example.
- Added a 'Theme' drop down to **F1** which allows a single switch for multiple color combos on the various game list entries.

- Added ability to launch a softlist machine on its own, e.g. 'mame.exe a2600', instead of going into its softlist view. Hold the **Shift** key down while hitting enter or double clicking on the machine. This is akin to starting a console with no cart in it.
- Reintroduced an **F1** toggle to hide non-working mechanical machines, removing 15,193 entries from the game list; this is now on by default.
- Added a drop down in **F1** for icon scale presets based on monitor resolutions and DPI combos. This controls the scaled size of the regular and large icon display on said monitors.
- **Alt-I** in the game list will cycle through the various icon scale presets.

NOTES

- In the 1080p 96DPI era, the initial IV/Play dimensions were 1015x432; a 2.35:1 aspect ratio in honor of the 1960's [TohoScope](#) tech used to make Godzilla movies. Today, on 4K monitors, the same ratio is used but at a default of 1523x648 (144dpi and 1.5X scaling).

VERSION HISTORY & DEVELOPMENT TIMELINE

Sunday, December 17, 2006	Rev. a	Concept Doc
Tuesday, June 26, 2007	Rev. b	Design Doc
Saturday, April 09, 2011	Rev. c	Design Doc
Sunday, May 26, 2011	1.0	User Guide
Sunday, June 12, 2011	1.5	User Guide
Sunday, October 16, 2011	1.5.3	User Guide
Wednesday, November 02, 2011	1.5.5	User Guide
Sunday, November 20, 2016	1.8.1.2	User Guide
Tuesday, January 15, 2019	1.8.1.2	User Guide
Saturday, April 17, 2021	1.8.4	User Guide
Friday, May 5, 2023	1.8.5	User Guide
Thursday, September 25, 2025	2.1	User Guide

APPENDIX

ARCHITECTURE

Application Architecture Summary

The IV/Play application is a high-performance MAME frontend for Windows. Its architecture is heavily optimized for fast startup, responsive filtering, and efficient rendering of large game lists by leveraging a deferred asset loading model, a DirectX-based rendering engine, a multi-layered caching strategy, and in-memory data processing.

Core Application Flow

The application's lifecycle begins in Program.cs, which ensures only a single instance is running and handles the initial detection of the MAME executable if it's not configured.

The `MainForm`` class serves as the central orchestrator. It features two distinct startup paths to maximize performance:

- **Cold Start (First Run / Database Rebuild):** On its first launch or after a MAME update is detected, the application performs a one-time data build. A splash screen is displayed while the `XmlParser` processes MAME's entire output and the `DatabaseManager` builds a local SQLite database. This is the longest startup path but is only required once per MAME version.
- **Warm Start ("Pure Speed" Path):** This is the normal startup path, optimized to render an interactive UI in under 500ms. It is divided into two stages:

Fast Path: Executes only the essential tasks required to display the GameList. It loads the main database into memory and populates the GameList from a high-speed binary cache. The UI is then immediately displayed.

Deferred Path: After the UI is interactive, heavier, non-essential tasks are run on background threads. These include loading the full icon atlas from the `AssetCache` and scanning for DAT files (`history.xml`, etc.). The UI displays placeholder icons, which are then smoothly faded into the final, high-resolution icons once the deferred loading is complete.

Data Management and Persistence

IV/Play's data strategy is robust, designed to transform MAME's raw XML output into a queryable format.

- **Data Source:** The primary source of game information is the XML output from the mame.exe -listxml command.
- **Parsing:** The XmlParser class launches a MAME process and deserializes the resulting XML stream directly into a list of Machine objects. This process populates complex, nested properties for each machine, such as CPU, sound, and display information.
- **Persistence:** A DatabaseManager using Entity Framework and SQLite stores the processed Machine data in a local database file (IV-Play.db). The database is completely rebuilt if the MAME executable has been updated or if the application's database schema has changed, ensuring data consistency.
- **In-Memory Processing:** For maximum filtering speed, the entire `Machines` table is loaded from the database into a static list in memory (`DatabaseManager.AllMachines`) at startup. This is a key component of the 'Fast Path' that enables a sub-500ms time-to-interactive on warm starts.

Rendering and UI

The user interface is built for performance, moving beyond standard Windows Forms controls for its core component.

- **DirectX Engine:** The GameList control is a custom UserControl that bypasses standard GDI+ rendering. It uses the Vortice library to interface directly with DirectX (Direct3D11 and Direct2D), enabling smooth scrolling and high-quality image scaling, transparency, and effects like scanlines.
- **Modular Control:** The GameList logic is separated into partial classes for drawing (GameList.Drawing.cs), user input (GameList.Input.cs), and navigation/data handling (GameList.Navigation.cs), keeping the code organized.
- **UI Management:** The MainForm manages the GameList control and auxiliary windows, such as the ConfigForm for settings and the FilterDialog for search queries.

Multi-Layered Caching Strategy

To achieve its high performance and rapid startup, the application employs several layers of caching.

- **Icon Atlas Cache:** AssetCache generates a single large PNG image (a "texture atlas") containing all game icons. This dramatically improves rendering performance by minimizing the number of individual images the GPU needs to manage. The atlas is cached to disk and validated using a hash of the source icon files, so it's only rebuilt when icons change. During a warm start, placeholder icons are rendered instantly while the full atlas is loaded from this cache on a background thread.
- **GameList Cache:** For the most common startup scenario (no active filters **or** custom game list), GameListCache saves the fully sorted and processed game lists to a compact binary file. Loading from this cache is significantly faster than querying the database and re-sorting thousands of entries.
- **Artwork Cache:** ArtCache implements a memory-based Least Recently Used (LRU) cache for game artwork like snapshots and flyers. This reduces disk reads for frequently viewed art. It uses a background ArtLoader to fetch images from disk asynchronously without blocking the UI.
- **DAT File Cache:** The InfoParser, used for history.xml and mameinfo.dat, creates a JSON-based cache of the parsed data. This cache is validated against the source file's timestamp, preventing redundant parsing of these large text files on subsequent runs.

Filtering and Search

The application features a sophisticated filtering system that allows for both simple and complex queries.

- **Natural Language Parser:** NaturalLanguageParser translates intuitive, human-readable queries (e.g., capcom fighting 1994) into the application's strict, SQL-like advanced filter syntax (e.g., manufacturer LIKE "%capcom%" AND description LIKE "%fighting%" AND year = "1994").
- **Advanced Filtering:** This advanced syntax supports a wide range of operators (=, >, LIKE, NOT, OR, AND, BETWEEN), parentheses for grouping, and specific keywords for fields like year, manufacturer, and driver.
- **In-Memory Execution:** The filter query is converted into a LINQ expression and executed directly against the in-memory list of all games (DatabaseManager.AllMachines), resulting in near-instantaneous filtering.

Configuration and Diagnostics

IV/Play includes tools for configuration and troubleshooting.

- **Settings Management:** A central, static SettingsManager class manages all user-configurable options. Settings are persisted to a text-based IV-Play.cfg file. The manager also handles the auto-detection of art folders and DAT files based on the MAME executable's location.
- **Diagnostics:** The application integrates a buffered Logger to record detailed events with minimal performance overhead by flushing writes to disk periodically rather than instantly. A StartupProfiler is also used to time key operations during the launch sequence, helping to identify and diagnose performance issues. It now reports both the primary "time-to-interactive" metric and a separate set of "Deferred Tasks (Post-UI)" timings in the log file.

FILTER AND SEARCH GUIDE

The filter bar, accessible with **Ctrl-F** or **Ctrl-E**, is a powerful tool for finding specific games. Simple words, special keyword modifiers, or complex, SQL-like advanced queries can be used to narrow down the game list.

Basic Searching

- The easiest way to search is to enter the desired terms.
 - Simple Text Search: Typing more than one word finds games that contain ALL of those words in their information. For example, pacman midway will find games that have both "pacman" AND "midway" in their data.
 - Exact Phrases: To search for an exact phrase, it should be enclosed in double quotes. For example, "metal slug" will find games with that exact phrase in their title.
-

Using Keyword Modifiers

Special keywords can be used to quickly filter the list, either alone or combined with other search terms.

Game Status & Attributes

- **Working Status:**
 - Working only: Shows only games marked as working.
 - Nonworking: Shows only games marked as not working.
 - Imperfect: Shows games with a driver status of "imperfect".
 - **Clones:**
 - No clones or parents only: Hides all clone games.
 - Clones only: Shows only clone games.
 - **Favorites:**
 - Favorites only or show my favorites: Shows only games from the favorites list.
 - Exclude favorites or not favorites: Hides all games from the favorites list.
 - **Hardware & Media Types:**
 - Mechanical: Shows only mechanical games.
 - Chd games: Shows games that use a hard drive or CD-ROM image.
 - Laserdisc games: Shows games that use a laserdisc.
 - Bios only: Shows only BIOS files.
 - No bios: Hides all BIOS files.
 - Vector: Shows games that use vector graphics.
 - Raster: Shows games that use raster (pixel-based) graphics.
 - Samples or no samples: Shows or hides games that require external sound sample files.
 - Lightgun or no lightgun: Shows or hides games that use a lightgun controller.
 - Multi screen or multi monitor: Shows games that use more than one monitor.
 - Screenless: Shows machines that have no display output.
 - **Display Orientation:**
 - Vertical: Shows games with a vertical or portrait monitor orientation.
 - Horizontal: Shows games with a horizontal or landscape monitor orientation.
-

Date & Decade Filtering

Natural language can be used to find games from a specific time period.

- **Specific Years & Ranges:**
 - Made in 1993
 - After 1990
 - Before 1985 (or pre-1985)
 - From 1980 to 1985
 - **Decades:**
 - 80s games or in the 90s (finds games from 1980-1989)
 - Late 90s (finds games from 1997-1999)
 - Early 90s (finds games from 1990-1993)
-

Special Diagnostic Filters

- **Working clones of nonworking parents:** Finds working clone games whose parent version is marked as non-working. This is useful for finding playable versions of games when the main set is broken.
 - **Nonworking parents with working clones (or False Parents):** Finds non-working parent games that have at least one working clone. This is useful for auditing game sets that could potentially be marked as working.
-

Advanced Searching

For more precise control, an advanced query syntax can be used. The parser will automatically use this mode if field-specific searches or operators like =, >, or LIKE are used.

Field-Specific Searches

Specific data fields can be targeted using a colon (:). The syntax is field:value.

- **Available Fields:**
 - Year (alias: y)
 - Manufacturer (alias: mfg)
 - Sourcefile (aliases: hardware, driver)
 - Name
 - Description (alias: desc)
 - **Field Negation (!):** A value can be excluded by putting an exclamation mark (!) before it.
 - Manufacturer:!Capcom shows games from every manufacturer EXCEPT Capcom.
 - Year:!1994 shows games from every year EXCEPT 1994.
-

The Power User's Toolbox

For maximum control, complex queries can be built using a full, SQL-like language.

Operator	Meaning	Example
=	Equality (string or number)	year = 1994
!=	Inequality	manufacturer != "Sega"
> <=	Numeric comparison	year > 1990
LIKE	Substring match (% is a wildcard)	description LIKE "%Fighter%"
BETWEEN	Inclusive numeric range	year BETWEEN 1990 AND 1992
IN	Matches any value in a list	manufacturer IN ("Taito", "Irem")
AND OR	Boolean logic	year > 1990 AND manufacturer = "Sega"
()	Groups expressions for order of operations	year > 1990 AND (manufacturer = "Sega" OR manufacturer = "Namco")

Combined Examples

Any of the above methods can be mixed and matched to create powerful, specific queries.

- **Working, non-clone Midway games:** midway working only no clones
- **Capcom games from the late 90s, excluding Street Fighter:** capcom late 90s description:!"Street Fighter"
- **All Sega or Namco games made after 1990:** year > 1990 AND (manufacturer = "Sega" OR manufacturer = "Namco")

CREDITS

Creator & Designer (2006-Present): John L. Hardy IV

Initial Development (2011-2016): Matan Bareket

Modernization & AI Pair-Programming (2025): Gemini / CoPilot / ChatGPT / Grok