WinFellow

v0.5.8

Generated by Doxygen 1.8.13

Contents

Chapter 1

WinFellow Source Code Concepts

Petter Schau, May 2000

This document contains **some** core concepts for the source code in Fellow v0.4.

It attempts to describe the overall structure and organization of the source-code.

Module Types

WinFellow is a modular design where there are basically 3 kinds of modules:

Emulation modules which emulate some part of the Amiga hardware. These modules are independent of, and free from any OS-specific code. The modules are in some cases heavily implemented with x86-assembly code.

Device wrapper modules which isolate the emulation modules from the device implementation modules. They provide functions to the emulation modules which does exactly what the emulation modules need. In turn, the device wrapper modules use OS-specific code in order to program the hardware for a specific OS.

Administration modules which perform the overall management of the emulator.

Standard Module API

Standard Events

Each module implements the following standard Fellow module functions to respond to various events:

- Initialization
- Shutdown
- Starting emulation
- Stop emulation
- Hard reset
- End of line
- End of frame

Module API / Configuration

Additionally there are module specific functions to perform the following functions:

- Configuration
- Program the associated hardware

Examples

Sound Device Module

For example, the sound device module implements most of the standard events. It has a collection of configuration functions in order to set the desired sound quality, and it has functions which allow the sound-emulation module to add new samples to the sound device. Lastly, there is a function which starts playback of a new buffer.

Graphics Device Module

Another example is the graphics device module, which also implements most standard events. Configuration involves setting the current graphics mode to one of the available modes. It has a function which delivers a frame-pointer to the emulation draw module, which enables the emulation to draw pixels on the graphics-device. Note that it does not matter to the draw-module whether the actual device is a window, full-screen or just a buffer in normal memory as long as the device wrapper module can provide a pointer to the image-buffer.

Minor / Major modules

In most cases, device-wrapper modules are owned and initialized by the emulation modules. The emulation draw module initializes a graphics device wrapper modules which it uses to perform the drawing.

Module List

In general, each emulation module corresponds to a pair of files (C part and assembly part). In one case, the emulation is so complex that the emulation process has been divided into 3 modules. (Graphics.)

Emulation Modules

- Blitter emulation module (blit.c / blita.s)
- Bus emulation module (bus.c / busa.s)
- Cia chips emulation module (cia.c / ciaa.s)
- Copper emulation module (copper.c / coppera.s)
- CPU emulation module (cpu.c / cpua.s)
- Drawing emulation module (draw.c / drawa.s)
- Filesystem mapping module (ffilesys.c and UAE specific files)
- Floppy emulation module (floppy.c)
- Hardfile emulation module (fhfile.c / fhfile.s)
- Gameport emulation module (gameport.c)
- Graphics emulation module (graph.c / grapha.s)
- Keyboard emulation module (kbd.c)
- Memory emulation module (fmem.c / fmema.s)
- Sound emulation module (sound.c / sounda.s)
- Sprite emulation module (sprite.c / spritea.s)
- Wav file sound output module (wav.c)

Device Wrapper Modules

- Graphics device module (gfxdrv.c)
- Joystick device module (joydrv.c)
- Keyboard device module (kbddrv.c / kbdparser.c)
- Mouse device module (mousedrv.c)
- Sound device module (sound.c)
- GUI module (wgui.c)

Administration Modules

- Fellow module (fellow.c)
- Configuration module (config.c)
- OS-specific startup module (winmain.c)

Modules Overview

This overview explains each module in terms of how it takes input, what makes the module do work, and what the output is. It also tries explain how it is related to other modules and how it cooperates with those modules.

Blitter Emulation Module Overview

The blitter module emulates the Amiga blitter in software. The blitter is basically a chip that reads data from one or more source memory locations, combines the data read using a logical expression and finally writes the result to a destination location. The blitter operates in 3 major modes, copy, line or fill.

On initialization, the module creates a lot of static lookup tables which is used during emulation of a particular mode.

The state of the blitter is contained in a number of "registers". These registers are accessed using memory-access functions registered in the memory emulation module. The memory module will call those functions whenever data is written to the accociated memory locations.

When the BLTSIZE register is written, a blit starts. (Once again happens indirectly through the memory module via. a memory access function.)

The blit is defined by the current state of the blitter registers. The state of the blitter is analyzed to decide what needs to be done. The time needed for the operation is calculated, ie. the time an Amiga blitter would have needed to complete the blit.

What happens next is that a "blitter-event" is added to the bus emulation module. This causes a function to be called a number of virtual clock ticks (which we calculated) into the future. Nothing else is done, and by returning, the emulator goes on to emulate something else.

Later, when the event we scheduled in the bus emulation module is due, code is run that emulates the blit. And an IRQ is raised which cause other things to happen which is not described here.

Bus Emulation Module Overview

The bus emulation module is the heart of the emulator. It contains a virtual tick counter, and a queue of events. Basically it is just one loop which does the following:

- Take event off the start of the queue.
- Set virtual tick counter to the time the next event happens.
- Run code associated with the event.

Everything emulated is run from this loop. Each module needs to schedule new events to keep running. Possible events are:

- CPU instruction event
- Copper instruction event
- End of line event
- Blitter event
- IRQ event
- CIA event
- End of frame event

At any time, there will always be at least two events in the queue, End of line and End of frame. Unless the CPU runs a STOP instruction, the CPU is also in the queue.

This model is chosen to reflect the fact that at any one time, there are many things going on in parallell inside an Amiga. It is more efficient and a lot cleaner that the different modules schedule their own tasks through this mechanism than the more obvious one where the main loop would have to ask each module in turn, do you want to do something now? Mostly, the answer would have been no, and a lot of wasted CPU-time.

The module also contains the handlers for End of line and End of frame, these two events do a lot of bookkeeping work for various modules (by means of calling EndOfFrame() and EndOfLine() functions, for instance, indirectly, these events drive the sound, floppy and graphics emulation.

CIA Chips Emulation Module Overview

The two Cia chips controls various peripherals, such as the floppy-drives and keyboard. They also contain a number of timers and some other features that are uncommon and not emulated.

Basically, the Cia module is driven by writes to the registers. (Memory access functions registered with the memory module.) In turn this causes functions in the floppy-emulation module to be called when floppy-related status registers are being written.

The timers are handled by always keeping the expiration time updated. This is turn affects the Cia event that is on the bus emulation queue. When the Cia event expires, the Cia event is called and there is no additional overhead to keep track of the timers.

The third aspect of Cia emulation is that the floppy and keyboard module will sometimes update their status registers located in the Cia module. Such as making a new keyboard scancode visible, or asking the Cia to generate an IRQ. (Cia IRQs are in turn scheduled through the CPU-emulation module.)

This module has no device wrapper.

Copper Emulation Module Overview

The copper is a simple processor that writes data one word at a time into custom chip registers. It can wait for a specified raster beam position and for the blitter. The copper is programmed by creating a list of instructions in memory. (Somewhat oversimplified view of the operation.)

The copper emulation is driven from one side by memory access functions registered in the memory emulation module. (Start and stop copper, define the memory location of the copper instruction list to name most.)

On the other side, the emulation of copper lists is event driven through the bus emulation module. The time for the next copper instruction is an event, and emulation code is called in the copper module. The copper module calls memory location access functions registered in the memory emulation module to write values to custom chip registers.

This module has no device wrapper.

CPU Emulation Module Overview

The CPU is the largest emulation module. It emulates the Motorola 68k CPU. The overall path of emulation of a CPU-instruction is:

- The bus emulation module runs a CPU-instruction event.
- The opcode-word is read using the memory-emulation module.
- The opcode-word is used as an index into a jump-table which contains a pointer to a function which emulates the opcode.
- The instruction is parsed, usually it includes reading one or more words using the memory-emulation module, do an operation on the data, and write the result to memory once again using the memory-emulation module. The internal state of the CPU is updated (flags, PC, stack etc.).
- The next CPU-instruction event is scheduled by calculating the clock ticks for this module.

The CPU module also handles the scheduling of IRQ events.

Other details of internal aspects of the CPU-emulation is omitted here. There are for instance a lot of pre-calculated data in tables in order to help parsing what to do for each opcode, and the case of handling IRQs, exceptions and maintaining the integrity of the status-register. There are also some optimizations to speed up reading and writing to memory, and a couple of optimization defects that are difficult to remove without replacing the module. (Evaluating EA twice in many cases and much faster flag handling.)

This module has no device wrapper.

Graphics, Sprite and Drawing Emulation Module Overview

The graphics handling is divided in three. The custom registers for graphics and sprites are held by the graphics and sprite modules. They are modified the usual way by registering memory location access functions with the memory module. In order to keep track of what happens to the screen and to figure out what to draw, much bookkeeping is done to calculate helper variables that describe the current screen properties in a cleaner way than the actual registers.

The overall process of rendering a line of the Amiga screen is as follows:

- At the end of each virtual line, the current state of the custom registers define the appearance of the line. The rendering is one line at the time, which does not capture some special effects accurately. The EndOfFrame() handler cause rendering to happen.
- The appearance of the line is rendered into a temporary buffer in a format that is preprocessed to aid fast drawing later. This includes translating the planar Amiga graphics to chunky pixels which describe the color for each pixel on the line. Calculations involve location of the horisontal borders, and possible hidden pixels that must be skipped, but not shown. Or the line might be a line that shows the vertical border. In any case, the temporary buffer contains a full Amiga line rendered as chunky pixels. There are also some flags to catch special cases, such as a line with no bitmap pixels, in that case the color of the entire line is remembered.
- Sprites are added to the temporary line. Similar calculations about location or absense of sprites are done to figure out where they are.
- For each virtual line we repeat this process, building a temporary buffer for the entire screen.
- When the entire frame is done, the EndOfFrame() handler cause the draw-module to do work.
- A framepointer to the current buffer (in case of double/triple buffering) is obtained from the graphics device wrapper module.
- Each line for the entire screen is processed one at the time. The temporary rendering provided by the graphics module is translated to the pixel-format on the host screen one pixel at the time. Blank lines are skipped if their colors are unchanged.

The draw module uses the graphics device wrapper to gain access to the screen.

Filesystem Mapping Module Overview

The filesystem module is provided from the UAE-project. It plugs itself into AmigaOS by making itself available as an expansion card. (Expansion cards are added using some functions in the memory emulation module.) When the expansion card is discovered by AmigaOS during boot-up, the "filesystems expansion-card ROM" is mapped into memory and an initialization routine is called. This routine creates descriptions for each drive mounted as a filesystem in the "ROM" tagged as resident modules. Later in the boot-process, AmigaOS will scan the ROM for resident module identifiers and initialize each drive for use with AmigaOS. The code in the filesystem is written in C (x86 native), but the ROM contains short stubs which execute illegal instructions which the emulator recognizes and in turn the correct routines for handling the entire filesystem is called.

That description leaves a lot desired, but on the overall, that’s how it operates.

This module has no device wrapper. It is implemented using standard C IO.

Hardfile Module Overview

The basic concepts for hooking the hardfiles into AmigaOS is the same as for the Filesystem mapping module. Using a virtual expansion card and resident module tags in a ROM area provided by the expansion card. The only difference is that native functions are not called by executing illegal instructions, but by writing certain values to predefined locations in the hardfile device ROM and letting the memory module call memory location access functions for those addresses.

This module has no device wrapper. It is implemented using standard C IO.

Floppy Emulation Module Overview

The floppy emulation module state is modified through writes to certain CIA registers, and some custom registers. Apart from handling inserted disk-images, such as keeping track of where the head is and whether the motor is started, the main operation in the emulation is reading and writing data located on the disk. Disks are emulated as files which is a complete (non-MFM encoded) image of a floppy disk.

Reading and writing starts when a dedicated custom register is written, the module keeps track of that the usual way by implementing a memory location access function. The current state for the drive defines the operation (read/write, track, side, motor, length of read). Floppy data is fed slowly into the Amiga memory to improve compatibility with loaders that do not expect data arriving at high speed. (Though it can be configured to do this fast.)

Words are read at the end of each virtual line using the EndOfLine() handler.

This module has no device wrapper. It is implemented using standard C IO.

Gameport Emulation Module Overview

The gameport emulation module is a collection of memory location access functions which responds to the custom registers allocated to the gameport.

Additionally it has variables that contains the current state of a joystick and mouse for each of the two ports, there are returned by the memory location access functions whenever they are called.

The gameport emulation module depends on device wrapper modules for mouse and joystick support. Each module must support asychronous notification of changes to the mouse or joystick state, and call functions in the gameport emulation module to change the current state for the mouse and joystick.

This module requires device wrapper modules for mouse and joystick.

Keyboard Emulation Module Overview

The keyboard emulation module maintains the current state of a virtual Amiga keyboard. There are three levels of symbolic key mappings for the virtual keyboard. The module itself keeps track of keypresses using a symbolic PC keymap. This PC keyboard is mapped onto actual Amiga keys on one side, allowing great freedom in choosing how to map the keys. On the other hand, there are the actual keycodes, which are OS-dependent and translated into symbolic PC-keys by the keyboard device wrapper module.

The module cooperates with the Cia emulation module. The current (or last) scancode is contained in a register in the Cia-module. The keyboard module writes the new scancode to the Cia when there is a new one (and the Cia will raise an IRQ for it.)

From the other side, the keyboard device wrapper module is assumed to receive asynchronous notification about keychanges on the real keyboard and call a function in the keyboard module to get it processed for emulation of the keypress.

In between there is a queue to buffer fast arriving keypresses, since there is a minimum of virtual ticks between each new scancode to arrive in the emulated Amiga.

This module requires device wrapper for the actual keyboard.

Memory Emulation Module Overview

The memory emulation module sets up a virtual memory space for use by the other emulated Amiga devices. At the core is description tables which define the entire memory space in terms of the following:

- Memory is divided up into 64K banks.
- Each bank is described by memory location access functions for reading and writing byte, word and long from the memory in the bank. There is an optional pointer to the actual memory for this bank, and a flag indicating whether executable code can exist in the bank.

The module provides functions for setting up chip, fast, bogo, expansion cards, Kickstart image and Custom chip register memory areas. Other modules can map themselves into memory using standard functions. In the case of register handling, a second level of memory location access functions are used to trap access to a specific register. The memory emulation module handles this for custom registers, the Cia sets this up by itself in its own bank access functions. This module has no device wrapper.

Sound and Wav Emulation Modules Overview

Sound is emulated through the use of a state-machine which is needed to accurately emulate all aspects of the Amiga sound hardware. (See HRM).

Sound is controlled through trapping access to custom chip registers through the memory module the usual way. In addition, actual sound output is produced by running the audio state machine a number of times in each EndOf↩ Line().

In order to play the sample values produced by the emulation, a sound device wrapper module is needed.

The Wav module is an alternate sound-device wrapper module, although it implements the sound device wrapper API, samples are not played, but written to a Wav file on disk.

More information on the sound module exists in a separate document.

This module requires device wrapper for a sound device.

Device Wrapper Module Overviews

There is no documentation for the device wrappers. Read the include files for each device wrapper to see the API, read comments in the source code and see how the device-independent modules make use of them to accomplish their tasks.

Chapter 2

HOWTO: Development environment setup

WinFellow is designed to be portable to different operating systems. However the project files contained in the Git repository are built using Microsoft Visual Studio.

Currently the following software should be used for development in the Git master branch:

- Visual Studio 2019

The community edition of Visual Studio 2019 can be used to compile WinFellow; it even features debugging and profiling.

For access to the Git repository, the `GitHub Desktop` client is required.

For basic contributions, the components mentioned above should be sufficient.

To be able to compile release builds using the automated build script, a number of additional components must be installed and added to the search path:

- Visual Studio 2019 with C++ desktop development components must be installed, including the following additions:
  - the legacy XP toolset components (v141_xp; this is described in a little more detail in `this article`.
  - due to a Visual Studio bug, CppUnitTest.h can sometimes not be located; as a workaround, the following command can be executed (see `here`):

  ```
  move "%ProgramFiles(x86)%\Microsoft Visual Studio\2019\Community\Common7\↩
  IDE\VC\VCTargets\*.props" "%ProgramFiles(x86)%\Microsoft Visual Studio\2019\↩
  Community\MSBuild\Microsoft\VC\v150"
  ```

- PowerShell execution policy must be set to unrestricted (both for the 32 as well as the 64 bit PowerShell)
- Git for Windows 64 Bit must be installed and added to the search path; usually Notepad++ is used as default editor
- the module posh-git must be installed
- the LyX Bundle including MikTeX must be installed and lyx.exe added to the search path
- 7-Zip - search path
- NSIS - search path
- Visual Studio Locator (vswhere.exe) must be located somewhere within the search path

WinFellow was ported from DOS Fellow, which was based in large parts on assembler code that has been converted to C in the master branch. The assembler based code still exists in the assembly_based branch. To work with that branch, NASM2 is needed additionally.

The WinFellow user manual was created and is being edited using LyX.

A documentation (work in progress) targeted specifically at developers can also be created using doxygen and the batch file "Build-Doxygen-Documentation.cmd" included in the source code archive; a TeX distribution like MikTeX or TeX Live must be installed for the generation to succeed.

Coding Style Guidelines

To ensure consistency across the different modules that are developed by different authors, we believe it is reasonable to have a common set of guidelines all developers follow.

- For optimum layout, configure the tab width to 8, and enable the replacing of tabs with spaces.
- Configure the indentations to 2.
- No line should be longer than 200 characters.
- Prefer the use of built-in C++ datatypes for new code (bool instead of BOOLE).
- New modules can be created in C++.
- Instead of one-line statements, the use of the curly brackets {} is preferred, but not mandatory.

Chapter 3

HOWTO: doxygen environment setup

Doxygen is a documentation system for C++, C, Java, and various other languages.

It can help you in three ways:

- It can generate an on-line documentation browser (in HTML) and/or an off-line reference manual (in LaTeX) from a set of documented source files. There is also support for generating output in RTF (MS-Word), PostScript, hyperlinked PDF, compressed HTML, and Unix man pages. The documentation is extracted directly from the sources, which makes it much easier to keep the documentation consistent with the source code.
- You can configure doxygen to extract the code structure from undocumented source files. This is very useful to quickly find your way in large source distributions. You can also visualize the relations between the various elements by means of include dependency graphs, inheritance diagrams, and collaboration diagrams, which are all generated automatically.
- You can also use doxygen for creating normal documentation (as I did for this manual).

Doxygen is developed under Linux and Mac OS X, but is set-up to be highly portable. As a result, it runs on most other Unix flavors as well. Furthermore, executables for Windows are available.

doxygen setup using Windows Subsystem for Linux

Since the compilation of the doxygen documentation using Windows has become error-prone and frustrating, involving components that are not necessarily well-maintained, an alternate method to compile it using the Windows Subsystem for Linux has been devised. Obviously, the same thing can be achieved on a native Ubuntu Linux system.

First, setup WSL on Windows by executing

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```

Reboot the system if required.

Then install Ubuntu from the Windows store: https://www.microsoft.com/store/p/ubuntu/9nblggh4msv6

When everything is set up and an initial username/password have been configured, install the required components by executing

```
sudo apt install doxygen graphviz texlive-latex-base texlive-latex-recommended texlive-latex-extra
```

Accept/confirm all prompts and wait for the packages to install.

cd into the WinFellow source code directory and find the doxygen subdirectory.

Execute `Build-Doxygen-Documentation.sh`

doxygen setup using Windows

The following setup files were used in the past to generate the WinFellow documentation on Windows 10. The newest version of doxygen where this could be tested successfully was 1.8.10.

- doxygen: doxygen-1.8.10.windows.x64.bin (http://www.stack.nl/~dimitri/doxygen/download.html)
- MiKTeX: basic MiKTeX installer (http://miktex.org/download)
- GhostScript: gs927w64.exe (https://www.ghostscript.com/download/gsdnld.html)
- Graphviz: graphviz-2.38.0.msi (https://graphviz.gitlab.io/_pages/Download/Download_windows.html)

1. Install doxygen using the default settings. Later versions than 1.8.10 have recently shown problematic; this needs to be examined in more detail.
2. Install MiKTeX using the default settings.
3. Install GhostScript (64 bit Windows version) using the default settings. Append the directory "C:\Program Files\gs\gs9.27\bin" to the PATH environment variable, separated by a semicolon.
4. Install Graphviz using the default settings. Confirm any UAC (user account control) prompts you might encounter.
5. Verify that the following files can be executed from the command-line before attempting to build the documentation (directories need to be included in PATH - if one of them does not work, try logging off and back on again):
   - doxygen.exe
   - latex.exe
   - pdflatex.exe
   - gswin64c.exe
   - dot.exe

The doxygen environment should now be properly configured to build the WinFellow documentation.

To build the documentation, enter the doxygen directory and execute the file Build-Doxygen-Documentation.cmd. Verify that the table of contents contains proper page numbers, and that the documentation contains graphs. If one of these is missing, then something is wrong with the installation that requires more troubleshooting.

The documentation should be found as WinFellow-doxygen.pdf within the doxygen directory.

Chapter 4

HOWTO: Preparing WinFellow for a new public release

This HOWTO describes the steps necessary to prepare a new public WinFellow release. This is to ensure that a uniform process is used that results in the same quality of release archives, regardless of who takes care of building and spreading the release. This HOWTO should always be updated as necessary.

- Make sure that there are no open issues left in the `issue tracker` that are assigned to be fixed in this release.
- Ensure that there are no local modifications/uncommitted changes in the local Git repository (compiling a release build should automatically enforce this).
- For compiling the new release, the build environment should be setup like recommended here (HOWTO: Development environment setup).
- Verify that no unnecessary debug/trace logs are written (floppy.log, capsdump.txt, ...)
- Using a debug build, verify that no memory leaks occur for basic as well as new functions; these would be logged in files called `WinFellowCrtMallocReport_WinFellowVersion_YYYYMMDD_hhmmss.log`. Ensure that all modules include code to support CRT malloc/C++ new debug logging, so that filenames and line numbers of the leaks are included.
- Perform regression tests according to the HOWTO: Regression Testing.
- The third digit of the version number should usually be increased by one (in the files versioninfo-wcrev.h and versioninfo-wcrev.rc, as well as in the doxyfile, the user guide and the NSIS installer file WinFellow.nsi). Make sure to also update the VS_VERSION_INFO structure in GUI_versioninfo-wcrev.rc.
- Update the ReadMe/FAQ to reflect the latest changes in this release.
- Compile the build using the script CompileReleaseBuild.ps1; this will take care of a number of things:
  - ensure that no local modifications exist, perform a Git update and compile a clean release build
  - copy .exe, .pdb file and PDF user guide into a folder named WinFellow_...
  - Generate ChangeLog; ensure that Git commits are included in the resulting file; copy the ChangeLog.txt into the WinFellow_... folder
  - copy GPL license terms (PDF) into the WinFellow_... folder
  - zip the folder using 7-Zip
  - clean up the source code directory, copy the GPL license terms into it and zip the folder using 7-Zip
- beta testing is usually performed before a public release; recently beta builds have been distributed via Dropbox; in the past we posted the betas to the EAB `private : WinFellow beta release` forum

- announce availability of the beta in the forum and to the fellow-beta mailing list
- If feedback is positive, upload the release with highlight information to GitHub and post announcements (public EAB support forum, fellow-announce, ...)
- update the hugo website with the release highlights (WinFellow branch 'hugo', insert a new post), recompile the hugo website and commit/push the resulting files to the gh-pages branch to update the public website

Unclarified points:

- Do we use profile-guided optimization, and if we do, how do we run it? Or do we use whole-program optimization instead?
- how do we ensure the beta exe is not leaked, can we mark it as beta somehow so that beta and release build differ?

Chapter 5

HOWTO: Regression Testing

Several issues were fixed over time in WinFellow's emulation core. To avoid them resurfacing in the future, the following tests should be performed on a regular basis, at the very least after major changes.

When testing large batches of titles, to speed up testing it is possible to make the emulation run at maximum speed by setting the configuration file option *sound_output=interrupts*; this can also be configured in Amiga Forever's override.ini, allowing quick testing of a group of RP9 packages.

Automated Test Cases

The following is a list of test cases where the excution can be automated using WinFellow's automation capabilities; most of these are emulator core bugfixes.

The actual individual test case result must be gathered by examining the resulting screenshots.

### r749: Changed the constant for lines in a frame

Test that the demo "Global Trash" by The Silents will proceed beyond the second screen and display the "Global Trash" logo. Before this commit, it would freeze before that appeared.

### r764: Flags for eorw and orw, calculation of flags fixed

Test that Micro Machines [cr CSL] can be started without issues/the main menu appears. Before this commit, it would crash after the cracktro.

### r814: removed filling in MFM encoding of floppy sectors

Test that Ballistix [cr Defjam] can be started without issues. Before this commit, it froze right after starting it up.

### r824: Changed how the CPU driven sound stub issues IRQs

Test that Ballistix [cr Defjam] can be started without issues. Before this commit, it would freeze before displaying the Psyclapse logo.

Other known titles affected: Bombuzal, Brides of Dracula, Dungeon Master, F-18 Interceptor, Fiendish Freddy, Grand Prix Circuit, Indiana Jones 3, Loom, Neuromancer, Personal Nightmare, Revenge of the Mutant Camels, Robocop, Test Drive, Test Drive 2, Vroom, Where in the world is Carmen Sandiego, Wings of Fury

### r831: Redesigned chipset interrupt handling

Test that the demo "Seeing is Believing" can be started without issues - before this commit, the session would freeze immediately upon start.

Other known titles affected: Championship Manager 95 Italy, Crystal Kingdom Dizzy, Flashback, Life & Death, Lionheart, Nitro, Pacland, Pipemania, Populous, Puffy's Saga, Speedball, Ugh!

### r832: fixed movep instruction bug, improved CPU exception handling

Test that the game Airborne Ranger can be started sucessfully, without graphical issues. Before this commit, the screen would look garbled.

Other known titles affected: Barbarian 1 (Psygnosis), Fire Force, KGB, Larry 3, Larry 5, Police Quest 2, The Lost Vikings.

### r865: Loaders using disksync different from 0x4489 and 0x8914 with standard ADF files

Test that Prince of Persia loads. It would not start earlier.

Test that Lemmings 2 (uncracked) loads successfully; before the fix, it would indefinitely read from the floppy disk while loading the main menu.

Comment in 0.4.4 source also suggests North and South is affected, though this could not be reproduced recently. This fix supersedes the fix in r843, which did not take 0x8914 into account.

### r873: floppy words per line should be 2 instead of 3

Test that the demo Guardian Dragon II by Kefrens loads successfully, and continues beyond the first Kefrens logo. The scrolling text between the two blue statues needs to appear. Before this change, the sound would start garbling and the emulation session would reboot.

### r888: sprites in hires dual playfield mode

Test that during the loader screen in Decaying Paradise by Andromeda, a blue rotating triangle-shaped logo is visible. Before this change, the triangle was invisible.

### r898: increased sprite action/merge item list sizes to a maximum of 5.000 entries

Start the 1MB chipmem version of State of the Art (Spaceballs) with only 512kB of chipmem configured. The demo will cause a crash of the emulator session, but the emulator should remain responsive. Before this change, a crash to the desktop would occur.

Update: This fix is replaced by Git f99a94de. "To be safe increase sprite max list items to 100" It follows other code changes that reduced the need for very large sprite list buffers. This is still a valid test. Also test the game Megalomania, it creates a larger sprite list than State of the Art when it crashes after the initial intro text when OCS is selected. (The actual crash is due to copy-protection.)

### r906: Sprite DMA was being disabled instead of waiting

Start Arkanoid, hit F1 and start in round 7 (move mouse to the right to select level). Verify that enemy sprites are coming down from the top of the screen. Before this change, they were not visible.

Arkanoid is updating vstart/vstop with the copper.

### r949: PC -2 saved on the stack for address exception

Start the game Double Dragon II [cr Oracle] and verify that it loads the main menu after exiting the cracktro by hitting Enter.

Before this change, a loader animation would be displayed indefinitely.

### r950: proper "short frame" when display is interlaced and frame is short

Start the game Project-X Special Edition '93 and enter the game. Before this change, there would be graphical corruption and the emulator would become unresponsive.

### r954: fix copper list load

Start the demo Sequential by Andromeda with automatic interlace compensation. The (interlaced) intro graphic must be displayed properly, without interlace flickering.

Before this change, it would have inverted lines.

Also affected by this is the game The Ninja Warriors; the game would not load before this change and can now be started.

If copper DMA was off during "end of frame", and is being turned on for the first time after that, it also loads the copper list pointer.

### r955: graph frame pointer NULL pointer exception

Start the game First Samurai and proceed through the intro. When prompted to insert disk 2 into DF0:, do not change the disk, just press fire. The game will crash, but the emulation session should remain open indefinitely.

Before this commit, the emulator would crash to the desktop with a NULL pointer exception.

### r957: clear DMA pending flag when blit is initiated from enabling DMA in wdmacon

Start the demo "Megademo 8" (Kefrens). Enter the "snake bite" section and verify that it loads normally.

Before this change, the emulator would hang in an endless loop and become unresponsive.

### r958: set floppy change bit high when no disk is selected

Start the game "Silkworm [cr Trilogy/t+4 Trilogy]" and verify that the cracktro can be left by pressing both mouse buttons simultaneously.

Before this commit, it was impossible to proceed beyond the cracktro.

Also known to be affected by this change is the game "Plan 9 From Outer Space", which would fail to proceed loading the second disk.

### r963: copjmp lost if triggered while dma was off and copper had already run to the end of its copper list

Start the demo Multica by Andromeda with automatic interlace compensation enabled. Verify that the initial intro screen featuring the Andromeda logo looks right.

Before this commit, it had an issue with inverted lines.

### r967: implemented chipmem / bogomem aliasing

Start the demo Wayfarer by Spaceballs with 512kB chipmem and no bogo memory. It should load normally, the emulator should not crash.

Before this commit, the session would crash. Related titles impacted by this change are Sensible Soccer 1.0, Cannon Fodder XMAS Edition and Toki.

### r969, r971: CIA timer fix

Verify in the game Atomix that the main game can be started.

Before this fix, it would hang when entering the game.

### r974: bit-field '020 instruction code reimplemented

Start the demo Lotus Esprit Turbo Challenge 96k by Scarab. It should start correctly.

Before this change it would fail to load with a black screen after the intro. I makes use of bit-field instructions during decrunching.

In the same commit, ASL overflow handling was improved, and a flag check regarding MULU was fixed; no test cases are known for these changes.

### Git 86bd011 and Git xxxxxx: Two disk related fixes and modification later

The game Amegas, packed into one file, does a disk access right after decrunching that requires the motor bit to be set in advance, and hung. This is a detail metioned in the HRM. Check that the game starts.

The game "The Games: Summer edition" stepped the disk head beyond the end of the disk and hung. It reads (with disksync) from track 80 where no data is. Max track set to 83 now, and data for upper tracks are random. (Will eventually generate sync.) Check that the game loads. Note that the game has broken intro graphics.

Outrun steps to track 80 and needs the extended max track limit. Check that Outrun loads.

### Git f99a94de: Last of several commits regarding "Sprites on HAM"

Verify that sprites on top of HAM resolution bitplane graphics works.

Two cases using this:

Fairlight - My Room demo A red filled vector cube (the sprite) is moving around on the screen, part of the middle screen is HAM resolution. Make sure the cube looks perfect wherever it moves.

Silents - Ice demo One of the first parts is a white screen with a picture of a warrior girl. The screen should contain a basic filled vector geometry object (the sprite) on top of the image moving around with no artifacts.

### Git 9a9ceb4: Added missing checks for disabled drives

Disabled floppy drives were not properly off-line with regard to some signals. This caused the game Winter Olympics 94 to endlessly scan for more drives. Check that the game loads.

Note: The game's scan is bugged, if you enable 4 drives, it will scan forever.

### Git 8b0c112: Re-encode track data to MFM from ADF after floppy-write

Supercars with bytebandit virus would not load because the virus re-wrote track 0. The internal MFM buffer's sector headers was out of order with what the game's trackloader expected, and it failed when pressing fire to start game. (Loads some first.) Emulator now regenerates MFM from disk for ADF's after writes to get the sector order.

Note: Supercars needs additional fix to work. See Git <TBD>.

Manual Test Cases

The following test cases must be executed manually as they have not been automated.

### r897: Alt+F4 in RetroPlatform mode when undo is enabled and a change was performed

Test that Alt+F4 in RetroPlatform mode will close the emulation session. Perform a write operation on a floppy where undo is enabled, and close the session using Alt+F4. The undo dialog must be usable both when clicking ok or cancel. Before this change, the emulation session would always be closed.

### r941: reset of RetroPlatform causes input devices to no longer function

Start the A500 system, click into the emulator window to capture the cursor. Hold Esc to release cursor and click the Reset button. After the reset, verify that the mouse is usable in Workbench.

### r972, r973: RetroPlatform escape key handling

Using default escape key ESC

1. verify that the cracktro of Cannon Fodder 2 [cr PDX] can be left by tipping ESC; holding ESC and releasing after the interval should have no effect (release input devices)
2. verify in Turrican II that the main game will be quit when tipping ESC; holding for the configured interval and releasing should have no effect (release input devices)
3. verify that pressing the key for longer than the configured interval releases the mouse cursor BEFORE releasing the key (end of frame handler)
4. verify in the game F-19 Stealth Fighter, that tipping the ESC key will delete a character from the roster; releasing after the configured interval should not have an effect, except to release the input devices
5. verify that using the escape key while emulation is paused will release the devices, but will not send the escape key when the session is resumed (Cannon Fodder 2 cracktro)

Configure escape key to A

1. verify in a Workbench CLI that tipping A will produce a single "a" on the screen; holding and releasing A should have no effect (release input devices)

### r987: clipping and scaling, screenshots in Amiga Forever

Edit a title to use PAL standard clipping and verify it is displayed correctly in 1x mode. Take a screenshot, verify it is ok.

Edit a title to use PAL (maximum) clipping and verify it is displayed correctly in 1x mode. Take a screenshot, verify it is ok.

Start Arkanoid and verify it is displayed correctly

- in 1x mode. Take a screenshot, verify it is ok.
- in 2x mode. Take a screenshot, verify it is ok.

Edit a title to use Automatic clipping and verify the maximum screen area is visible in 1x mode. Take a screenshot, verify it is ok.

Edit a title to use custom clipping of a small area and verify it is displayed correctly

- in 1x mode. Take a screenshot, verify it is ok.
- in 2x mode. Take a screenshot, verify it is ok.

### Git 78ff087 (and other previous commits)

Start a standalone WinFellow session, configure it for DirectDraw mode. Configure scaling to 1x and and load Workbench. Hit <PrntScrn> to take a screenshot and ensure it is being saved to the pictures folder. Reconfigure to 2x, 3x and 4x as well as auto scaling and save a screenshot each time. Reconfigure the WinFellow session for Direct3D mode with 1x scaling and take another screenshot. Do the same for 2, 3x, 4x and auto scaling. Ensure the screenshots are saved properly and that they look like expected. Test this both for reconfiguring a running session, as well as starting a new emulation session/restarting the emulator.

Start Amiga Forever. Configure the WinFellow plugin for DirectDraw mode (Tools->Options->Emulation, Plugins, WinFellow, Graphics API). Start the Amiga 1000 system; click the 1x scaling button and close the emulation session. Restart the emulation session (it should still be in 1x mode upon start), wait for it to boot, escape the mouse cursor and start the clipping editor (Tools->Edit Screen Clip). Ensure that the whole Amiga screen buffer is visible with a blue background in the clipping editor, there should be no missing black areas. Cancel the clipping editor dialog. Save a screenshot to the pictures folder via the screenshot button and verify its creation. Do the same in 2x, 3x and 4x modes and ensure the screenshots are saved correctly; use the clipping editor accordingly. Close the emulation window, reopen it and take screenshots again.

Configure the WinFellow plugin for Direct3D mode (Tools->Options->Emulation, Plugins, WinFellow, Graphics A↩ PI). Start the Amiga 1000 system; click the 1x scaling button and close the emulation session. Restart the emulation session (it should still be in 1x mode upon start), wait for it to boot, escape the mouse cursor and start the clipping editor (Tools->Edit Screen Clip). Ensure that the whole Amiga screen buffer is visible with a blue background in the clipping editor, there should be no missing black areas. Cancel the clipping editor dialog. Save a screenshot to the pictures folder via the screenshot button and verify its creation. Do the same in 2x, 3x and 4x modes and ensure the screenshots are saved correctly; use the clipping editor accordingly. Close the emulation window, reopen it and take screenshots again.

Chapter 6

Todo List

File FLOPPY.C

CAPS has been renamed to SPS, and a 64 bit version is available; update to a current version

enhance timing for flakey image support

Chapter 7

Hierarchical Index

7.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 8

Data Structure Index

8.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 9

File Index

9.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 10

Data Structure Documentation

10.1 _chipset_information Struct Reference

Data Fields

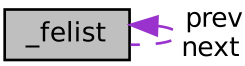
- bool **ecs**
- ULO **ptr_mask**
- ULO **address_mask**

The documentation for this struct was generated from the following file:

- chipset.h

10.2 _felist Struct Reference

Collaboration diagram for _felist:



Data Fields

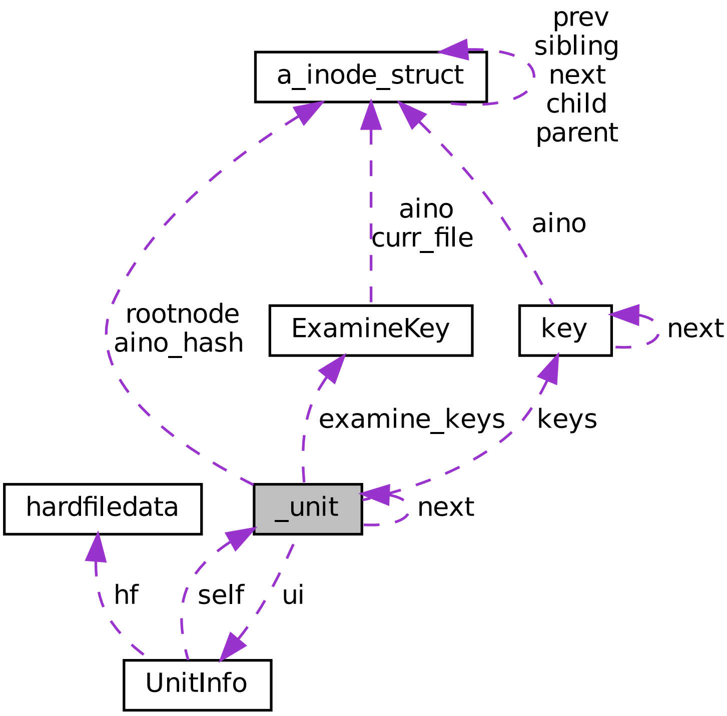
- struct _felist ∗ **next**
- struct _felist ∗ **prev**
- void ∗ **node**

The documentation for this struct was generated from the following file:

- LISTTREE.H

10.3 _unit Struct Reference

Collaboration diagram for _unit:



Data Fields

- struct _unit ∗ **next**
- uaecptr **dosbase**
- uaecptr **volume**
- uaecptr **port**
- uaecptr **locklist**
- uae_s32 **unit**
- UnitInfo **ui**
- char **tmpbuf3** [256]
- uaecptr **dummy_message**
- volatile unsigned int **cmds_sent**
- volatile unsigned int **cmds_complete**
- volatile unsigned int **cmds_acked**
- ExamineKey **examine_keys** [EXKEYS]
- int **next_exkey**
- unsigned long **total_locked_ainos**
- struct key ∗ **keys**
- uae_u32 **key_uniq**
- uae_u32 **a_uniq**

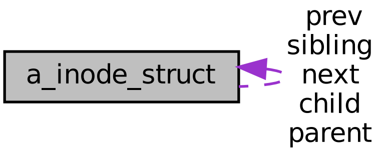
- a_inode **rootnode**
- unsigned long **aino_cache_size**
- a_inode ∗ **aino_hash** [MAX_AINO_HASH]
- unsigned long **nr_cache_hits**
- unsigned long **nr_cache_lookups**

The documentation for this struct was generated from the following file:

- FILESYS.C

10.4 a_inode_struct Struct Reference

Collaboration diagram for a_inode_struct:



Data Fields

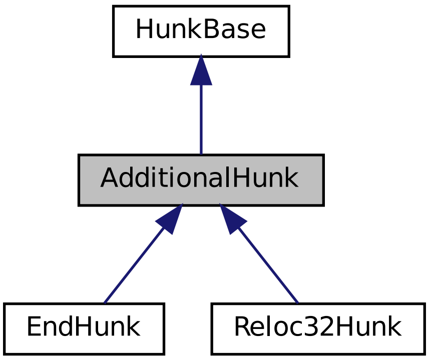
- struct a_inode_struct ∗ **next**
- struct a_inode_struct ∗ **prev**
- struct a_inode_struct ∗ **parent**
- struct a_inode_struct ∗ **child**
- struct a_inode_struct ∗ **sibling**
- char ∗ **aname**
- char ∗ **nname**
- char ∗ **comment**
- int **amigaos_mode**
- uae_u32 **uniq**
- unsigned long **locked_children**
- unsigned long **exnext_count**
- int **shlock**
- long **db_offset**
- unsigned int **dir**:1
- unsigned int **elock**:1
- unsigned int **has_dbentry**:1
- unsigned int **needs_dbentry**:1
- unsigned int **dirty**:1
- unsigned int **deleted**:1

The documentation for this struct was generated from the following file:

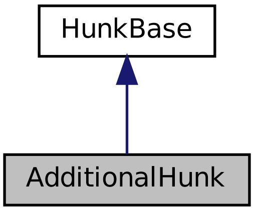
- fsdb.h

10.5 AdditionalHunk Class Reference

Inheritance diagram for AdditionalHunk:



Collaboration diagram for AdditionalHunk:



Public Member Functions

- ULO **GetSourceHunkIndex** ()
- void **Parse** (RawDataReader &rawReader) override=0
- **AdditionalHunk** (ULO sourceHunkIndex)

Private Attributes

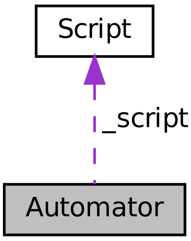
- ULO **_sourceHunkIndex**

The documentation for this class was generated from the following files:

- AdditionalHunk.h
- AdditionalHunk.cpp

10.6 Automator Class Reference

Collaboration diagram for Automator:



Public Member Functions

- void **RecordKey** (UBY keyCode)
- void **RecordMouse** (gameport_inputs mousedev, LON x, LON y, BOOLE button1, BOOLE button2, BOOLE button3)
- void **RecordJoystick** (gameport_inputs joydev, BOOLE left, BOOLE up, BOOLE right, BOOLE down, BO↩ OLE button1, BOOLE button2)
- void **RecordEmulatorAction** (kbd_event action)
- void **EndOfLine** ()
- void **EndOfFrame** ()
- void **Startup** ()
- void **Shutdown** ()

Data Fields

- string **ScriptFilename**
- bool **RecordScript**
- string **SnapshotDirectory**
- int **SnapshotFrequency**
- bool **SnapshotEnable**

Private Member Functions

- void **TakeSnapshot** ()

Private Attributes

- Script **_script**
- int **_snapshotsTaken** = 0
- int **_snapshotCounter** = 0

The documentation for this class was generated from the following files:

- Automator.h
- Automator.cpp

10.7 bit_field< FB, FE > Struct Template Reference

Static Public Member Functions

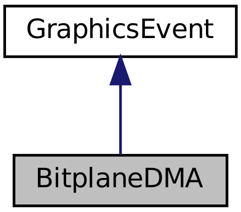
- static uint32 **mask** ()
- static bool **test** (uint32 value)
- static uint32 **extract** (uint32 value)
- static void **insert** (uint32 &data, uint32 value)

The documentation for this struct was generated from the following file:

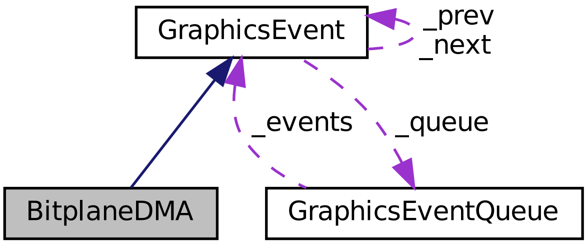
- m68k-tester.cpp

10.8 BitplaneDMA Class Reference

Inheritance diagram for BitplaneDMA:



Collaboration diagram for BitplaneDMA:



Public Member Functions

- void **Start** (ULO cycle)
- void **Stop** (void)
- virtual void **InitializeEvent** (GraphicsEventQueue ∗queue)
- virtual void **Handler** (ULO rasterY, ULO cylinder)
- void **EndOfFrame** (void)

Private Member Functions

- void **Log** (ULO line, ULO cylinder)
- UWO **ReadWord** (ULO address)
- void **IncreaseBplPt** (ULO ∗bplpt, ULO size)
- UWO **GetHold** (ULO bplNo, ULO bplsEnabled, ULO ∗bplpt)
- void **AddModulo** (void)
- void **SetState** (BPLDMAStates newState, ULO cycle)
- void **SetStateNone** (void)
- void **Restart** (bool ddfIsActive)
- void **FetchLores** (void)
- void **FetchHires** (void)

Private Attributes

- BPLDMAStates **_state**
- bool **_stopDDF**
- bool **_hasBeenActive**

Additional Inherited Members

The documentation for this class was generated from the following files:

- BitplaneDMA.h
- BitplaneDMA.c

10.9 BitplaneDraw Class Reference

Public Member Functions

- void **DrawBatch** (ULO rasterY, ULO rasterX)
- void **TmpFrame** (ULO next_line_offset)

Private Member Functions

- void **TempLores** (ULO rasterY, ULO pixel_index, ULO pixel_count)
- void **TempLoresDual** (ULO rasterY, ULO pixel_index, ULO pixel_count)
- void **TempLoresHam** (ULO rasterY, ULO pixel_index, ULO pixel_count)
- void **TempHires** (ULO rasterY, ULO pixel_index, ULO pixel_count)
- void **TempHiresDual** (ULO rasterY, ULO pixel_index, ULO pixel_count)
- void **TempNothing** (ULO rasterY, ULO pixel_index, ULO pixel_count)

Private Attributes

- ULO(∗ **_tmpframe**)[1024]

The documentation for this class was generated from the following files:

- BitplaneDraw.h
- BitplaneDraw.c

10.10 BitplaneUtility Class Reference

Static Public Member Functions

- static bool **IsLores** (void)
- static bool **IsHires** (void)
- static bool **IsDualPlayfield** (void)
- static bool **IsHam** (void)
- static bool **IsPlayfield1Pri** (void)
- static bool **IsDMAEnabled** (void)
- static ULO **GetEnabledBitplaneCount** (void)

The documentation for this class was generated from the following file:

- BitplaneUtility.h

10.11 blitter_state_ Struct Reference

Data Fields

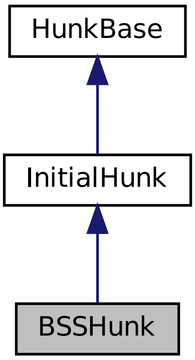
- ULO **bltcon**
- ULO **bltafwm**
- ULO **bltalwm**
- ULO **bltapt**
- ULO **bltbpt**
- ULO **bltcpt**
- ULO **bltdpt**
- ULO **bltamod**
- ULO **bltbmod**
- ULO **bltcmod**
- ULO **bltdmod**
- ULO **bltadat**
- ULO **bltbdat**
- ULO **bltbdat_original**
- ULO **bltcdat**
- ULO **bltzero**
- ULO **height**
- ULO **width**
- ULO **a_shift_asc**
- ULO **a_shift_desc**
- ULO **b_shift_asc**
- ULO **b_shift_desc**
- BOOLE **started**
- BOOLE **dma_pending**
- ULO **cycle_length**
- ULO **cycle_free**

The documentation for this struct was generated from the following file:

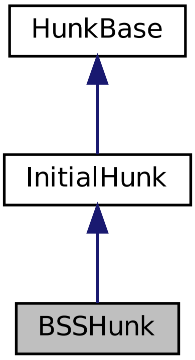
- BLIT.C

10.12 BSSHunk Class Reference

Inheritance diagram for BSSHunk:



Collaboration diagram for BSSHunk:



Public Member Functions

- ULO **GetID** () override
- void **Parse** (RawDataReader &rawDataReader) override
- **BSSHunk** (ULO allocateSizeInLongwords)

Static Private Attributes

- static const ULO **ID** = BSSHunkID

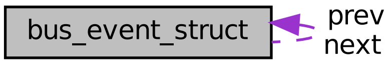
Additional Inherited Members

The documentation for this class was generated from the following files:

- BSSHunk.h
- BSSHunk.cpp

10.13 bus_event_struct Struct Reference

Collaboration diagram for bus_event_struct:



Data Fields

- struct bus_event_struct ∗ **next**
- struct bus_event_struct ∗ **prev**
- ULO **cycle**
- ULO **priority**
- busEventHandler **handler**

The documentation for this struct was generated from the following file:

- BUS.H

10.14 bus_screen_limits_ Struct Reference

Data Fields

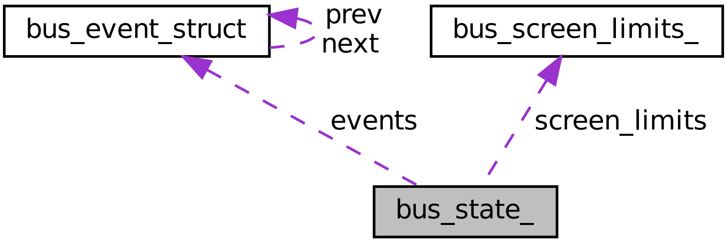
- ULO **cycles_in_this_line**
- ULO **cycles_in_this_frame**
- ULO **lines_in_this_frame**
- ULO **max_cycles_in_line**
- ULO **max_lines_in_frame**

The documentation for this struct was generated from the following file:

- BUS.H

10.15 bus_state_ Struct Reference

Collaboration diagram for bus_state_:



Data Fields

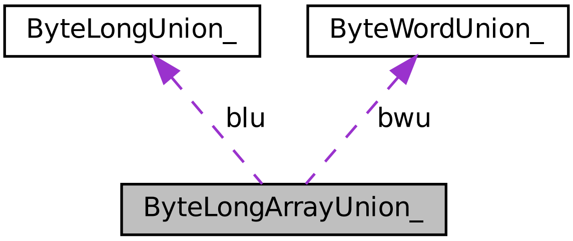
- ULL **frame_no**
- ULO **cycle**
- bus_screen_limits ∗ **screen_limits**
- bus_event ∗ **events**

The documentation for this struct was generated from the following file:

- BUS.H

10.16 ByteLongArrayUnion_ Union Reference

Collaboration diagram for ByteLongArrayUnion_:



Data Fields

- UBY **barray** [1024]
- ByteWordUnion **bwu** [512]
- ByteLongUnion **blu** [256]

The documentation for this union was generated from the following file:

- Planar2ChunkyDecoder.h

10.17 ByteLongUnion_ Union Reference

Data Fields

- ULO **l**
- UBY **b** [4]

The documentation for this union was generated from the following file:

- BitplaneUtility.h

10.18 ByteWordUnion_ Union Reference

Data Fields

- UWO **w**
- UBY **b** [2]

The documentation for this union was generated from the following file:

- BitplaneUtility.h

10.19 CapsDateTimeExt Struct Reference

Data Fields

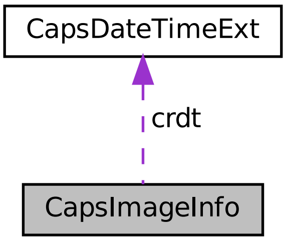
- UDWORD **year**
- UDWORD **month**
- UDWORD **day**
- UDWORD **hour**
- UDWORD **min**
- UDWORD **sec**
- UDWORD **tick**

The documentation for this struct was generated from the following file:

- CapsAPI.h

10.20 CapsImageInfo Struct Reference

Collaboration diagram for CapsImageInfo:



Data Fields

- UDWORD **type**
- UDWORD **release**
- UDWORD **revision**
- UDWORD **mincylinder**
- UDWORD **maxcylinder**
- UDWORD **minhead**
- UDWORD **maxhead**
- struct CapsDateTimeExt **crdt**
- UDWORD **platform** [CAPS_MAXPLATFORM]

The documentation for this struct was generated from the following file:

- CapsAPI.h

10.21 CapsTrackInfo Struct Reference

Data Fields

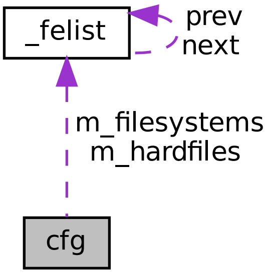
- UDWORD **type**
- UDWORD **cylinder**
- UDWORD **head**
- UDWORD **sectorcnt**
- UDWORD **sectorsize**
- UDWORD **trackcnt**
- PUBYTE **trackbuf**
- UDWORD **tracklen**
- PUBYTE **trackdata** [CAPS_MTRS]
- UDWORD **tracksize** [CAPS_MTRS]
- UDWORD **timelen**
- PUDWORD **timebuf**

The documentation for this struct was generated from the following file:

- CapsAPI.h

10.22 cfg Struct Reference

Collaboration diagram for cfg:



Data Fields

- ULO **m_configfileversion**
- STR **m_description** [CFG_FILENAME_LENGTH]
- STR **m_diskimage** [4][CFG_FILENAME_LENGTH]
- BOOLE **m_diskenabled** [4]
- BOOLE **m_diskreadonly** [4]
- BOOLE **m_diskfast**
- STR **m_lastuseddiskdir** [CFG_FILENAME_LENGTH]
- ULO **m_chipsize**
- ULO **m_fastsize**
- ULO **m_bogosize**
- STR **m_kickimage** [CFG_FILENAME_LENGTH]
- STR **m_kickimage_ext** [CFG_FILENAME_LENGTH]
- STR **m_kickdescription** [CFG_FILENAME_LENGTH]
- ULO **m_kickcrc32**
- STR **m_key** [CFG_FILENAME_LENGTH]
- bool **m_useautoconfig**
- bool **m_rtc**
- ULO **m_screenwidth**
- ULO **m_screenheight**
- ULO **m_screencolorbits**
- ULO **m_screenrefresh**
- bool **m_screenwindowed**
- bool **m_screendrawleds**
- ULO **m_frameskipratio**
- ULO **m_clipleft**
- ULO **m_cliptop**
- ULO **m_clipright**
- ULO **m_clipbottom**
- DISPLAYSCALE **m_displayscale**
- DISPLAYSCALE_STRATEGY **m_displayscalestrategy**
- bool **m_deinterlace**

- bool **m_measurespeed**
- DISPLAYDRIVER **m_displaydriver**
- GRAPHICSEMULATIONMODE **m_graphicsemulationmode**
- BOOLE **m_use_multiple_graphical_buffers**
- sound_emulations **m_soundemulation**
- sound_rates **m_soundrate**
- bool **m_soundstereo**
- bool **m_sound16bits**
- sound_filters **m_soundfilter**
- ULO **m_soundvolume**
- BOOLE **m_soundWAVdump**
- sound_notifications **m_notification**
- ULO **m_bufferlength**
- cpu_integration_models **m_CPUtype**
- ULO **m_CPUspeed**
- BOOLE **m_blitterfast**
- bool **m_ECS**
- felist ∗ **m_hardfiles**
- felist ∗ **m_filesystems**
- BOOLE **m_automount_drives**
- gameport_inputs **m_gameport** [2]
- bool **m_useGUI**
- BOOLE **m_config_applied_once**
- BOOLE **m_config_changed_since_save**

The documentation for this struct was generated from the following file:

- CONFIG.H

10.23 cfg_filesys Struct Reference

Data Fields

- STR **volumename** [64]
- STR **rootpath** [CFG_FILENAME_LENGTH]
- BOOLE **readonly**

The documentation for this struct was generated from the following file:

- CONFIG.H

10.24 cfg_hardfile Struct Reference

Data Fields

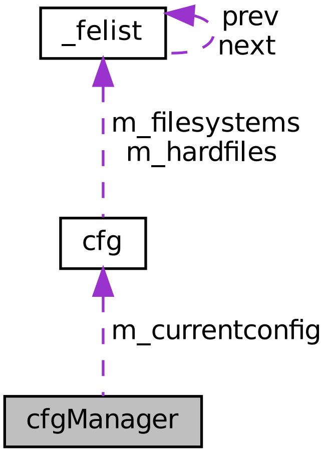
- STR **filename** [CFG_FILENAME_LENGTH]
- BOOLE **readonly**
- ULO **bytespersector**
- ULO **sectorspertrack**
- ULO **surfaces**
- ULO **reservedblocks**
- bool **hasrdb**

The documentation for this struct was generated from the following file:

- CONFIG.H

10.25 cfgManager Struct Reference

Collaboration diagram for cfgManager:



Data Fields

- cfg ∗ **m_currentconfig**

The documentation for this struct was generated from the following file:

- CONFIG.H

10.26 cia_state_ Struct Reference

Data Fields

- ULO **ta**
- ULO **tb**
- ULO **ta_rem**
- ULO **tb_rem**
- ULO **talatch**
- ULO **tblatch**
- LON **taleft**
- LON **tbleft**
- ULO **evalarm**
- ULO **evlatch**
- ULO **evlatching**
- ULO **evwritelatch**
- ULO **evwritelatching**

- ULO **evalarmlatch**
- ULO **evalarmlatching**
- ULO **ev**
- UBY **icrreq**
- UBY **icrmsk**
- UBY **cra**
- UBY **crb**
- UBY **pra**
- UBY **prb**
- UBY **ddra**
- UBY **ddrb**
- UBY **sp**

The documentation for this struct was generated from the following file:

- CIA.C

10.27 code Struct Reference

Data Fields

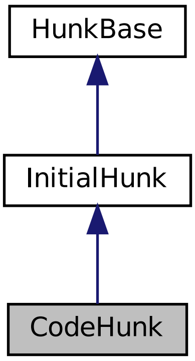
- unsigned char **op**
- unsigned char **bits**
- unsigned short **val**

The documentation for this struct was generated from the following file:

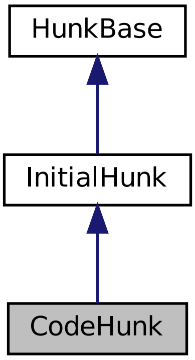
- inftrees.h

10.28 CodeHunk Class Reference

Inheritance diagram for CodeHunk:



Collaboration diagram for CodeHunk:



Public Member Functions

- ULO **GetID** () override
- void **Parse** (RawDataReader &rawDataReader) override
- **CodeHunk** (ULO allocateSizeInLongwords)

Static Private Attributes

- static const ULO **ID** = CodeHunkID

Additional Inherited Members

The documentation for this class was generated from the following files:

- CodeHunk.h
- CodeHunk.cpp

10.29 config_s Struct Reference

Data Fields

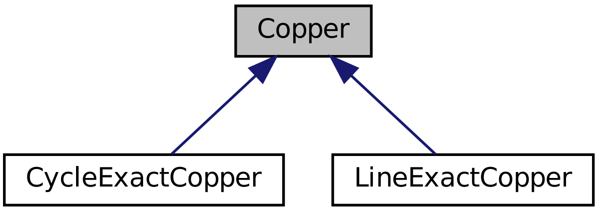
- ush **good_length**
- ush **max_lazy**
- ush **nice_length**
- ush **max_chain**
- compress_func **func**

The documentation for this struct was generated from the following file:

- deflate.c

10.30 Copper Class Reference

Inheritance diagram for Copper:



Public Member Functions

- virtual void **NotifyDMAEnableChanged** (bool new_dma_enable_state)=0
- virtual void **NotifyCop1lcChanged** ()=0
- virtual void **Load** (ULO new_copper_pc)=0
- virtual void **EventHandler** ()=0
- virtual void **EndOfFrame** ()=0
- virtual void **HardReset** ()=0
- virtual void **EmulationStart** ()=0
- virtual void **EmulationStop** ()=0

The documentation for this class was generated from the following file:

- COPPER.H

10.31 CopperRegisters Class Reference

Public Member Functions

- void **InstallIOHandlers** ()
- void **ClearState** ()
- void **LoadState** (FILE ∗F)
- void **SaveState** (FILE ∗F)

Data Fields

- ULO **copcon**
- ULO **cop1lc**
- ULO **cop2lc**
- ULO **copper_pc**
- bool **copper_dma**
- ULO **copper_suspended_wait**

The documentation for this class was generated from the following files:

- CopperRegisters.h
- CopperRegisters.cpp

10.32 cpu_data Struct Reference

Data Fields

- char **name** [32]
- int **data** [3]
- int **dis_func_no**

The documentation for this struct was generated from the following file:

- 68kgenerate.c

10.33 cpu_data_struct Struct Reference

Data Fields

- cpuInstructionFunction **instruction_func**
- ULO **data** [3]

The documentation for this struct was generated from the following file:

- CpuModule_Data.h

10.34 cpu_instruction_info Struct Reference

Data Fields

- char **instruction_name** [32]
- char **cpu_model_mask** [32]
- char **description** [32]
- char **function** [32]
- char **format** [32]
- char **size** [32]
- char **opcode** [32]
- char **eamask** [32]
- char **eamask2** [32]
- char **eaisdest** [32]
- char **reg** [32]
- char **disasm_func_no** [10]

The documentation for this struct was generated from the following file:

- 68kgenerate.c

10.35 cpuBfData Struct Reference

Data Fields

- LON **offset**
- ULO **width**
- ULO **normalized_offset**
- ULO **base_address**
- LON **base_address_byte_offset**
- ULO **base_address_byte_count**
- ULO **field**
- ULL **field_mask**
- ULO **dn**
- ULL **field_memory**

The documentation for this struct was generated from the following file:

- CpuModule_Instructions.c

10.36 ct_data_s Struct Reference

Data Fields

-
  ```
  union {
    ush freq
    ush code
  } fc
  ```

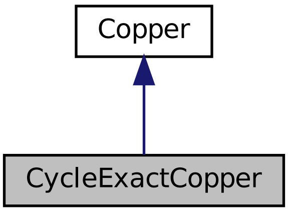
-
  ```
  union {
    ush dad
    ush len
  } dl
  ```

The documentation for this struct was generated from the following file:

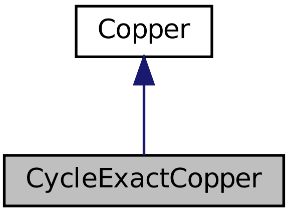
- deflate.h

10.37 CycleExactCopper Class Reference

Inheritance diagram for CycleExactCopper:



Collaboration diagram for CycleExactCopper:



Public Member Functions

- virtual void **NotifyDMAEnableChanged** (bool new_dma_enable_state)
- virtual void **NotifyCop1lcChanged** ()
- virtual void **Load** (ULO new_copper_pc)
- virtual void **EventHandler** ()
- virtual void **EndOfFrame** ()
- virtual void **HardReset** ()
- virtual void **EmulationStart** ()
- virtual void **EmulationStop** ()

Private Member Functions

- UWO **ReadWord** ()
- void **IncreasePtr** ()
- void **SetState** (CopperStates newState, ULO cycle)
- void **SetStateNone** ()

- bool **IsRegisterAllowed** (ULO regno)
- void **Move** ()
- void **Wait** ()
- void **Skip** ()
- bool **IsMove** ()
- bool **IsWait** ()
- void **ReadFirstWord** ()
- void **ReadSecondWord** ()
- void **TransferSecondWord** ()

Private Attributes

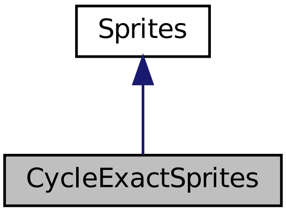
- CopperStates **_state**
- UWO **_first**
- UWO **_second**
- bool **_skip_next**

The documentation for this class was generated from the following files:

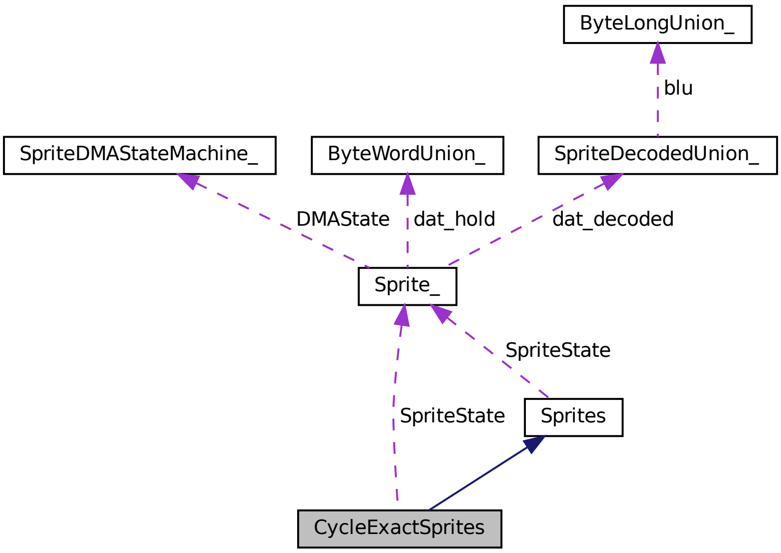
- CycleExactCopper.h
- CycleExactCopper.cpp

10.38 CycleExactSprites Class Reference

Inheritance diagram for CycleExactSprites:



Collaboration diagram for CycleExactSprites:



Public Member Functions

- virtual void **NotifySprpthChanged** (UWO data, unsigned int sprite_number)
- virtual void **NotifySprptlChanged** (UWO data, unsigned int sprite_number)
- virtual void **NotifySprposChanged** (UWO data, unsigned int sprite_number)
- virtual void **NotifySprctlChanged** (UWO data, unsigned int sprite_number)
- virtual void **NotifySprdataChanged** (UWO data, unsigned int sprite_number)
- virtual void **NotifySprdatbChanged** (UWO data, unsigned int sprite_number)
- void **OutputSprites** (ULO startCylinder, ULO cylinderCount)
- virtual void **HardReset** ()
- virtual void **EndOfLine** (ULO rasterY)
- virtual void **EndOfFrame** ()
- virtual void **EmulationStart** ()
- virtual void **EmulationStop** ()

Private Member Functions

- void **Arm** (ULO spriteNo)
- void **MergeLores** (ULO spriteNo, ULO source_pixel_index, ULO pixel_index, ULO pixel_count)
- void **MergeHires** (ULO spriteNo, ULO source_pixel_index, ULO pixel_index, ULO pixel_count)
- void **MergeHam** (ULO spriteNo, ULO source_pixel_index, ULO pixel_index, ULO pixel_count)
- void **Merge** (ULO spriteNo, ULO source_pixel_index, ULO pixel_index, ULO pixel_count)
- bool **InRange** (ULO spriteNo, ULO startCylinder, ULO cylinderCount)
- UWO **ReadWord** (ULO spriteNo)
- void **ReadControlWords** (ULO spriteNo)
- void **ReadDataWords** (ULO spriteNo)
- bool **IsFirstLine** (ULO spriteNo, ULO rasterY)
- bool **IsAboveFirstLine** (ULO spriteNo, ULO rasterY)

- bool **IsLastLine** (ULO spriteNo, ULO rasterY)
- bool **Is16Color** (ULO spriteNo)
- void **DMAReadControl** (ULO spriteNo, ULO rasterY)
- void **DMAReadData** (ULO spriteNo, ULO rasterY)
- void **DMAWaitingForFirstLine** (ULO spriteNo, ULO rasterY)
- void **DMAHandler** (ULO rasterY)
- void **ClearState** ()
- void **OutputSprite** (ULO spriteNo, ULO startCylinder, ULO cylinderCount)

Private Attributes

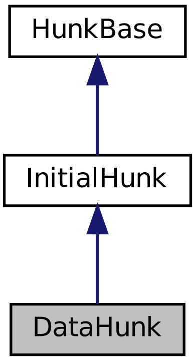
- Sprite **SpriteState** [8]

The documentation for this class was generated from the following files:

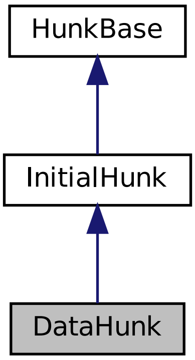
- CycleExactSprites.h
- CycleExactSprites.cpp

10.39 DataHunk Class Reference

Inheritance diagram for DataHunk:



Collaboration diagram for DataHunk:



Public Member Functions

- ULO **GetID** () override
- void **Parse** (RawDataReader &rawDataReader) override
- **DataHunk** (ULO allocateSizeInLongwords)

Static Private Attributes

- static const ULO **ID** = DataHunkID

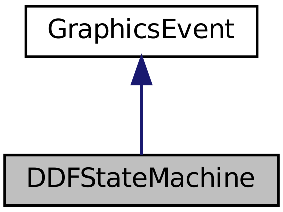
Additional Inherited Members

The documentation for this class was generated from the following files:

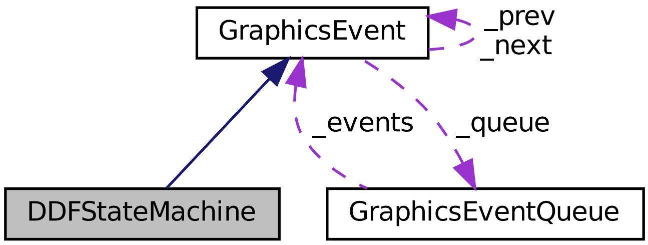
- DataHunk.h
- DataHunk.cpp

10.40 DDFStateMachine Class Reference

Inheritance diagram for DDFStateMachine:



Collaboration diagram for DDFStateMachine:



Public Member Functions

- bool **CanRead** (void)
- void **ChangedValue** (void)
- virtual void **InitializeEvent** (GraphicsEventQueue ∗queue)
- virtual void **Handler** (ULO rasterY, ULO cylinder)
- void **SoftReset** (void)
- void **HardReset** (void)
- void **EndOfFrame** (void)
- void **EmulationStart** (void)
- void **EmulationStop** (void)
- void **Startup** (void)
- void **Shutdown** (void)

Private Member Functions

- void **Log** (ULO line, ULO cylinder)
- ULO **GetStartPosition** (void)
- ULO **GetStopPosition** (void)
- ULO **GetFetchSize** (void)
- void **SetState** (DDFStates newState, ULO arriveTime)
- void **SetStateWaitingForFirstFetch** (ULO rasterY, ULO cylinder)
- void **SetStateWaitingForNextFetch** (ULO rasterY, ULO cylinder)
- void **DoStateWaitingForFirstFetch** (ULO rasterY, ULO cylinder)
- void **DoStateWaitingForNextFetch** (ULO rasterY, ULO cylinder)

Private Attributes

- DDFStates **_state**
- ULO **_minValidX**
- ULO **_maxValidX**

Additional Inherited Members

The documentation for this class was generated from the following files:

- DDFStateMachine.h
- DDFStateMachine.c

10.41 direct Struct Reference

Data Fields

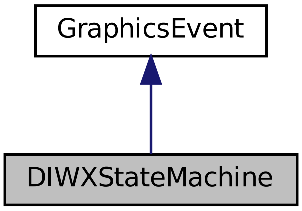
- char **d_name** [1]

The documentation for this struct was generated from the following file:

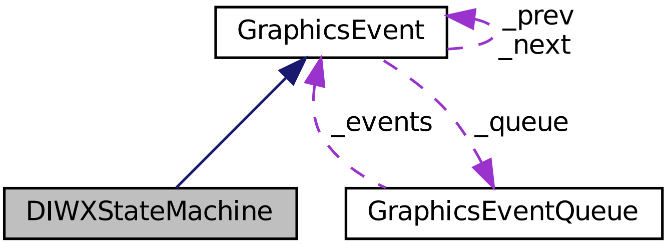
- UAE2FELL.H

10.42 DIWXStateMachine Class Reference

Inheritance diagram for DIWXStateMachine:



Collaboration diagram for DIWXStateMachine:



Public Member Functions

- bool **IsVisible** (void)
- void **ChangedValue** (void)
- virtual void **InitializeEvent** (GraphicsEventQueue ∗queue)
- virtual void **Handler** (ULO rasterY, ULO cylinder)
- void **SoftReset** (void)
- void **HardReset** (void)
- void **EndOfFrame** (void)
- void **EmulationStart** (void)
- void **EmulationStop** (void)
- void **Startup** (void)
- void **Shutdown** (void)

Private Member Functions

- void **Log** (ULO line, ULO cylinder)
- ULO **GetStartPosition** (void)
- ULO **GetStopPosition** (void)
- void **SetState** (DIWXStates newState, ULO arriveTime)
- void **SetStateWaitingForStartPos** (ULO rasterY, ULO cylinder)
- void **SetStateWaitingForStopPos** (ULO rasterY, ULO cylinder)
- void **DoStateWaitingForStartPos** (ULO rasterY, ULO cylinder)
- void **DoStateWaitingForStopPos** (ULO rasterY, ULO cylinder)
- void **OutputCylindersUntilPreviousCylinder** (ULO rasterY, ULO cylinder)

Private Attributes

- DIWXStates **_state**
- ULO **_maxValidX**

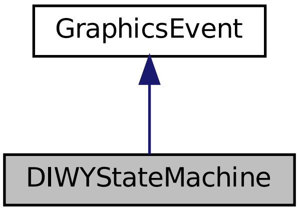
Additional Inherited Members

The documentation for this class was generated from the following files:

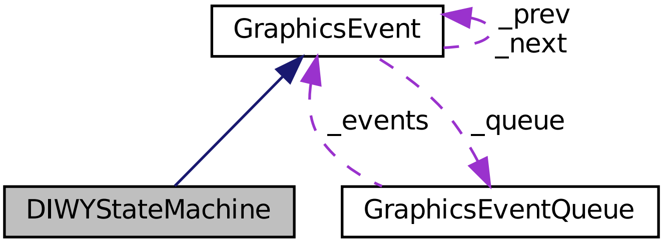
- DIWXStateMachine.h
- DIWXStateMachine.c

10.43 DIWYStateMachine Class Reference

Inheritance diagram for DIWYStateMachine:



Collaboration diagram for DIWYStateMachine:



Public Member Functions

- bool **IsVisible** (void)
- void **ChangedValue** (void)
- virtual void **InitializeEvent** (GraphicsEventQueue ∗queue)
- virtual void **Handler** (ULO rasterY, ULO cylinder)
- void **SoftReset** (void)
- void **HardReset** (void)
- void **EndOfFrame** (void)
- void **EmulationStart** (void)
- void **EmulationStop** (void)
- void **Startup** (void)
- void **Shutdown** (void)

Private Member Functions

- void **Log** (ULO line, ULO cylinder)
- ULO **GetStartLine** (void)
- ULO **GetStopLine** (void)
- void **SetState** (DIWYStates newState, ULO arriveTime)
- void **SetStateWaitingForStartLine** (ULO rasterY)
- void **SetStateWaitingForStopLine** (ULO rasterY)
- void **DoStateWaitingForStartLine** (ULO rasterY)
- void **DoStateWaitingForStopLine** (ULO rasterY)

Private Attributes

- DIWYStates **_state**
- ULO **_minValidY**
- ULO **_maxValidY**

Additional Inherited Members

The documentation for this class was generated from the following files:

- DIWYStateMachine.h
- DIWYStateMachine.c

10.44 draw_buffer_information Struct Reference

Data Fields

- UBY ∗ **top_ptr**
- UBY ∗ **current_ptr**
- ULO **width**
- ULO **height**
- ULO **pitch**
- ULO **bits**
- ULO **redsize**
- ULO **redpos**
- ULO **greensize**
- ULO **greenpos**
- ULO **bluesize**
- ULO **bluepos**

The documentation for this struct was generated from the following file:

- DRAW.H

10.45 draw_interlace_status Struct Reference

Data Fields

- bool **frame_is_interlaced**
- bool **frame_is_long**
- bool **enable_deinterlace**
- bool **use_interlaced_rendering**

The documentation for this struct was generated from the following file:

- draw_interlace_control.cpp

10.46 draw_mode Struct Reference

Data Fields

- ULO **id**
- ULO **width**
- ULO **height**
- ULO **bits**
- ULO **refresh**
- STR **name** [80]

The documentation for this struct was generated from the following file:

- DRAW.H

10.47 draw_rect Struct Reference

Public Member Functions

- ULO **GetWidth** () const
- ULO **GetHeight** () const
- **draw_rect** (ULO clip_left, ULO clip_top, ULO clip_right, ULO clip_bottom)

Data Fields

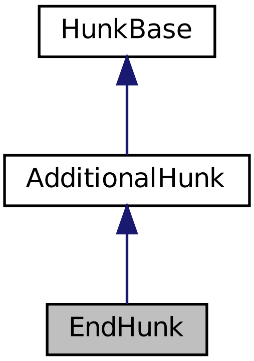
- ULO **left**
- ULO **top**
- ULO **right**
- ULO **bottom**

The documentation for this struct was generated from the following file:

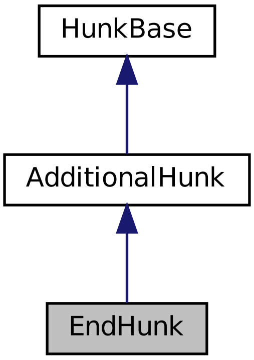
- DRAW.H

10.48 EndHunk Class Reference

Inheritance diagram for EndHunk:



Collaboration diagram for EndHunk:



Public Member Functions

- ULO **GetID** () override
- void **Parse** (RawDataReader &rawDataReader) override

Static Private Attributes

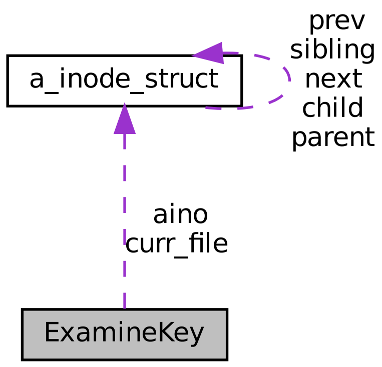
- static const ULO **ID** = EndHunkID

The documentation for this class was generated from the following files:

- EndHunk.h
- EndHunk.cpp

10.49 ExamineKey Struct Reference

Collaboration diagram for ExamineKey:



Data Fields

- uae_u32 **uniq**
- a_inode ∗ **aino**
- a_inode ∗ **curr_file**

The documentation for this struct was generated from the following file:

- FILESYS.C

10.50 ffilesys_dev Struct Reference

Data Fields

- STR **volumename** [FFILESYS_MAX_VOLUMENAME]
- STR **rootpath** [CFG_FILENAME_LENGTH]
- BOOLE **readonly**
- ffilesys_status **status**

The documentation for this struct was generated from the following file:

- FFILESYS.H

10.51 FileImage Class Reference

Public Member Functions

- void **SetHeader** (HeaderHunk ∗header)
- HeaderHunk ∗ **GetHeader** ()
- InitialHunk ∗ **GetInitialHunk** (ULO hunkIndex)
- void **AddInitialHunk** (InitialHunk ∗hunk)
- ULO **GetInitialHunkCount** ()
- AdditionalHunk ∗ **GetAdditionalHunk** (ULO hunkIndex)
- void **AddAdditionalHunk** (AdditionalHunk ∗hunk)
- ULO **GetAdditionalHunkCount** ()
- void **Clear** ()

Private Attributes

- std::unique_ptr< HeaderHunk > **_header**
- std::vector< std::unique_ptr< InitialHunk > > **_initialHunks**
- std::vector< std::unique_ptr< AdditionalHunk > > **_additionalHunks**

The documentation for this class was generated from the following files:

- FileImage.h
- FileImage.cpp

10.52 floppyDMAinfostruct Struct Reference

Data Fields

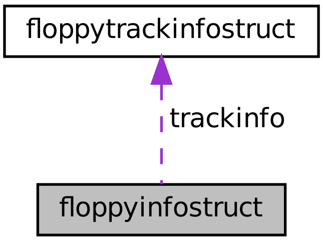
- ULO **dskpt**
- ULO **wordsleft**
- ULO **wait**
- BOOLE **wait_for_sync**
- BOOLE **sync_found**
- BOOLE **dont_use_gap**

The documentation for this struct was generated from the following file:

- FLOPPY.H

10.53 floppyinfostruct Struct Reference

Collaboration diagram for floppyinfostruct:



Data Fields

- FILE ∗ **F**
- ULO **tracks**
- BOOLE **zipped**
- ULO **compress_serno**
- BOOLE **sel**
- ULO **track**
- BOOLE **writeprot**
- BOOLE **dir**
- BOOLE **motor**
- BOOLE **side**
- BOOLE **step**
- BOOLE **enabled**
- BOOLE **changed**
- BOOLE **idmode**
- BOOLE **inserted**
- ULO **motor_ticks**
- ULO **insertedframe**
- ULO **idcount**
- UBY ∗ **mfm_data**
- floppytrackinfostruct **trackinfo** [FLOPPY_TRACKS]
- FLOPPY_STATUS_CODE **imagestatus**
- ULO **imageerror**
- STR **imagename** [CFG_FILENAME_LENGTH]
- STR **imagenamereal** [CFG_FILENAME_LENGTH]
- BOOLE **flakey**
- ULO ∗ **timebuf**

The documentation for this struct was generated from the following file:

- FLOPPY.H

10.54 floppytrackinfostruct Struct Reference

Data Fields

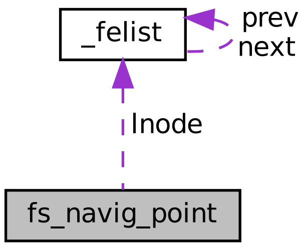
- ULO **file_offset**
- ULO **mfm_length**
- UBY ∗ **mfm_data**

The documentation for this struct was generated from the following file:

- FLOPPY.H

10.55 fs_navig_point Struct Reference

Collaboration diagram for fs_navig_point:



Data Fields

- UBY **drive**
- STR **name** [FS_WRAP_MAX_PATH_LENGTH]
- BOOLE **relative**
- BOOLE **writeable**
- ULO **size**
- fs_navig_file_types **type**
- felist ∗ **lnode**

The documentation for this struct was generated from the following file:

- FSNAVIG.H

10.56 fs_usage Struct Reference

Data Fields

- long **fsu_blocks**
- long **fsu_bfree**
- long **fsu_bavail**
- long **fsu_files**
- long **fsu_ffree**

The documentation for this struct was generated from the following file:

- FSUSAGE.H

10.57 fs_wrapper_point Struct Reference

Data Fields

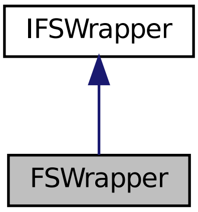
- UBY **drive**
- std::string **name**
- bool **relative**
- bool **writeable**
- ULO **size**
- fs_wrapper_file_types **type**

The documentation for this struct was generated from the following file:

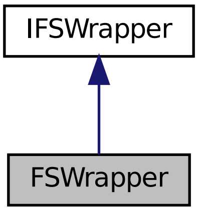
- IFSWrapper.h

10.58 FSWrapper Class Reference

Inheritance diagram for FSWrapper:



Collaboration diagram for FSWrapper:



Public Member Functions

- fellow::api::service::fs_wrapper_point ∗ **MakePoint** (const STR ∗point) override

Private Member Functions

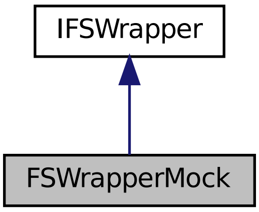
- fellow::api::service::fs_wrapper_file_types **MapFileType** (fs_navig_file_types type)

The documentation for this class was generated from the following files:

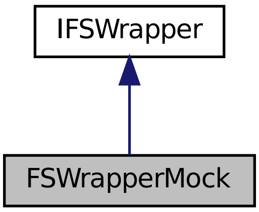
- FSWrapper.h
- FSWrapper.cpp

10.59 FSWrapperMock Class Reference

Inheritance diagram for FSWrapperMock:



Collaboration diagram for FSWrapperMock:



Public Member Functions

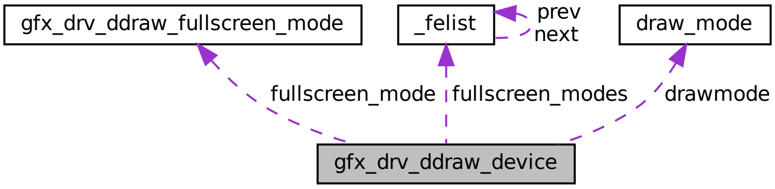
- ::fellow::api::service::fs_wrapper_point ∗ **MakePoint** (const STR ∗point) override

The documentation for this class was generated from the following files:

- FSWrapperMock.h
- FSWrapperMock.cpp

10.60 gfx_drv_ddraw_device Struct Reference

Collaboration diagram for gfx_drv_ddraw_device:



Data Fields

- LPGUID **lpGUID**
- LPSTR **lpDriverDescription**
- LPSTR **lpDriverName**
- LPDIRECTDRAW **lpDD**
- LPDIRECTDRAW2 **lpDD2**
- LPDIRECTDRAWSURFACE lpDDSPrimary

- LPDIRECTDRAWSURFACE lpDDSBack
- LPDIRECTDRAWSURFACE lpDDSSecondary
- DDSURFACEDESC **ddsdPrimary**
- DDSURFACEDESC **ddsdBack**
- DDSURFACEDESC **ddsdSecondary**
- LPDIRECTDRAWCLIPPER **lpDDClipper**
- felist ∗ **fullscreen_modes**
- gfx_drv_ddraw_fullscreen_mode ∗ **fullscreen_mode**
- ULO **buffercount**
- ULO **maxbuffercount**
- RECT **hwnd_clientrect_screen**
- RECT **hwnd_clientrect_win**
- draw_mode ∗ **drawmode**
- bool **use_blitter**
- bool **can_stretch_y**
- bool **no_dd_hardware**
- bool **windowed**

10.60.1 Field Documentation

10.60.1.1 lpDDSBack

```
LPDIRECTDRAWSURFACE lpDDSBack
```

Current backbuffer for Primary

10.60.1.2 lpDDSPrimary

```
LPDIRECTDRAWSURFACE lpDDSPrimary
```

Primary display surface

10.60.1.3 lpDDSSecondary

```
LPDIRECTDRAWSURFACE lpDDSSecondary
```

Source surface in blitmode

The documentation for this struct was generated from the following file:

- gfxdrv_directdraw.cpp

10.61 gfx_drv_ddraw_fullscreen_mode Struct Reference

Data Fields

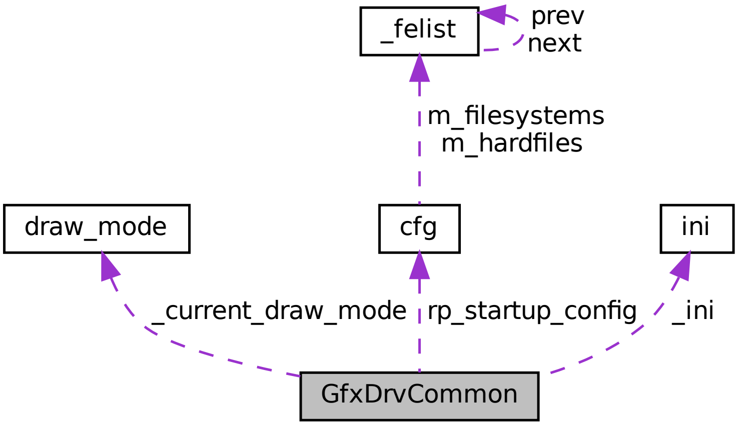
- ULO **width**
- ULO **height**
- ULO **depth**
- ULO **refresh**
- ULO **redpos**
- ULO **redsize**
- ULO **greenpos**
- ULO **greensize**
- ULO **bluepos**
- ULO **bluesize**
- ULO **pitch**

The documentation for this struct was generated from the following file:

- gfxdrv_directdraw.cpp

10.62 GfxDrvCommon Class Reference

Collaboration diagram for GfxDrvCommon:



Public Member Functions

- unsigned int **GetOutputWidth** ()
- unsigned int **GetOutputHeight** ()
- bool **GetOutputWindowed** ()
- void **SizeChanged** (unsigned int width, unsigned int height)
- void **DelayFlipTimerCallback** (ULO timeMilliseconds)
- bool **InitializeRunEvent** ()
- void **ReleaseRunEvent** ()
- void **RunEventSet** ()
- void **RunEventReset** ()
- void **RunEventWait** ()
- void **EvaluateRunEventStatus** ()
- void **NotifyDirectInputDevicesAboutActiveState** (bool active)
- bool **InitializeWindowClass** ()
- void **ReleaseWindowClass** ()
- void DisplayWindow ()
- void **HideWindow** ()
- bool **InitializeWindow** ()
- void **ReleaseWindow** ()
- LRESULT EmulationWindowProcedure (HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
- HWND **GetHWND** ()
- void **SetDrawMode** (draw_mode ∗dm, bool windowed)
- draw_mode ∗ **GetDrawMode** ()
- void **Flip** ()
- bool **EmulationStart** ()
- void **EmulationStartPost** ()
- void **EmulationStop** ()
- bool **Startup** ()
- void **Shutdown** ()

Data Fields

- bool **_displaychange**
- cfg ∗ **rp_startup_config**

Private Member Functions

- void **MaybeDelayFlip** ()
- void **DelayFlipWait** (int milliseconds)
- void **RememberFlipTime** ()
- int **GetTimeSinceLastFlip** ()
- void **InitializeDelayFlipTimerCallback** ()
- void **InitializeDelayFlipEvent** ()
- void **ReleaseDelayFlipEvent** ()

Private Attributes

- HANDLE **_run_event**
- HWND **_hwnd**
- volatile bool **_syskey_down**
- volatile bool **_win_active**
- volatile bool **_win_active_original**
- volatile bool **_win_minimized_original**
- draw_mode ∗ **_current_draw_mode**
- ini ∗ **_ini**
- unsigned int **_output_width**
- unsigned int **_output_height**
- bool **_output_windowed**
- int **_frametime_target**
- int **_previous_flip_time**
- volatile int **_time**
- volatile int **_wait_for_time**
- HANDLE **_delay_flip_event**

10.62.1 Member Function Documentation

10.62.1.1 DisplayWindow()

```
void DisplayWindow ( )
```

Show window hosting the amiga display.

Called on every emulation startup. In RetroPlatform mode, the player will take care of showing the emulator's window.

10.62.1.2 EmulationWindowProcedure()

```
LRESULT EmulationWindowProcedure (
            HWND hWnd,
            UINT message,
            WPARAM wParam,
            LPARAM lParam )
```

Window procedure for the emulation window.

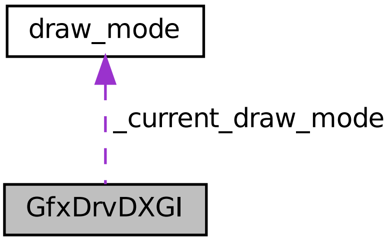
Distributes events to mouse and keyboard drivers as well.

The documentation for this class was generated from the following files:

- GfxDrvCommon.h
- GfxDrvCommon.cpp

10.63 GfxDrvDXGI Class Reference

Collaboration diagram for GfxDrvDXGI:



Public Member Functions

- void **ClearCurrentBuffer** ()
- void **SetMode** (draw_mode ∗dm, bool windowed)
- void **SizeChanged** (unsigned int width, unsigned int height)
- void **PositionChanged** ()
- void **NotifyActiveStatus** (bool active)
- bool **EmulationStart** (unsigned int maxbuffercount)
- unsigned int **EmulationStartPost** ()
- void **EmulationStop** ()
- bool **Startup** ()
- void **Shutdown** ()
- unsigned char ∗ **ValidateBufferPointer** ()
- void **InvalidateBufferPointer** ()
- void **GetBufferInformation** (draw_buffer_information ∗buffer_information)
- void **Flip** ()
- bool **SaveScreenshot** (const bool, const STR ∗)

Static Public Member Functions

- static bool **ValidateRequirements** ()

Private Member Functions

- bool **CreateAdapterList** ()
- void **DeleteAdapterList** ()
- void **RegisterMode** (unsigned int id, unsigned int width, unsigned int height, unsigned int refreshRate=60)
- void **RegisterModes** ()
- void **AddFullScreenModes** ()
- bool **CreateD3D11Device** ()
- void **DeleteD3D11Device** ()
- void **DeleteDXGIFactory** ()

- void **DeleteImmediateContext** ()
- bool **CreateSwapChain** ()
- void **DeleteSwapChain** ()
- bool **InitiateSwitchToFullScreen** ()
- void **ResizeSwapChainBuffers** ()
- void **SetViewport** ()
- DXGI_MODE_DESC ∗ **GetDXGIMode** (unsigned int id)
- bool **CreateAmigaScreenTexture** ()
- void **DeleteAmigaScreenTexture** ()
- ID3D11Texture2D ∗ **GetCurrentAmigaScreenTexture** ()
- bool **RenderAmigaScreenToBackBuffer** ()
- void **FlipTexture** ()
- const char ∗ **GetFeatureLevelString** (D3D_FEATURE_LEVEL featureLevel)
- bool **CreatePixelShader** ()
- void **DeletePixelShader** ()
- bool **CreateVertexShader** ()
- void **DeleteVertexShader** ()
- bool **CreateVertexAndIndexBuffers** ()
- void **DeleteVertexAndIndexBuffers** ()
- bool **CreateDepthDisabledStencil** ()
- void **DeleteDepthDisabledStencil** ()
- bool **SetShaderParameters** (const XMMATRIX &worldMatrix, const XMMATRIX &viewMatrix, const XM↩ MATRIX &projectionMatrix)
- void **CalculateDestinationRectangle** (ULO output_width, ULO output_height, float &dstHalfWidth, float &dstHalfHeight)
- void **CalculateSourceRectangle** (float &srcLeft, float &srcTop, float &srcRight, float &srcBottom)

Private Attributes

- GfxDrvDXGIAdapterList ∗ **_adapters**
- ID3D11Device ∗ **_d3d11device**
- ID3D11DeviceContext ∗ **_immediateContext**
- IDXGIFactory ∗ **_dxgiFactory**
- IDXGISwapChain ∗ **_swapChain**
- ID3D11VertexShader ∗ **_vertexShader**
- ID3D11PixelShader ∗ **_pixelShader**
- ID3D11Buffer ∗ **_vertexBuffer**
- ID3D11InputLayout ∗ **_polygonLayout**
- ID3D11Buffer ∗ **_indexBuffer**
- ID3D11Buffer ∗ **_matrixBuffer**
- ID3D11Texture2D ∗ **_shaderInputTexture**
- ID3D11Texture2D ∗ **_amigaScreenTexture** [AmigaScreenTextureCount]
- ID3D11DepthStencilState ∗ **_depthDisabledStencil**
- ID3D11SamplerState ∗ **_samplerState**
- unsigned int **_amigaScreenTextureCount**
- unsigned int **_currentAmigaScreenTexture**
- draw_mode ∗ **_current_draw_mode**
- bool **_resize_swapchain_buffers**

Static Private Attributes

- static bool **_requirementsValidated** = false
- static bool **_requirementsValidationResult** = false

The documentation for this class was generated from the following files:

- GfxDrvDXGI.h
- GfxDrvDXGI.cpp

10.64 GfxDrvDXGIAdapter Class Reference

Public Member Functions

- const GfxDrvDXGIOutputList & **GetOutputs** ()
- **GfxDrvDXGIAdapter** (IDXGIAdapter ∗adapter)

Private Member Functions

- void **EnumerateOutputs** (IDXGIAdapter ∗adapter)
- void **LogCapabilities** (IDXGIAdapter ∗adapter)

Private Attributes

- char **_name** [255]
- GfxDrvDXGIOutputList **_outputs**

The documentation for this class was generated from the following files:

- GfxDrvDXGIAdapter.h
- GfxDrvDXGIAdapter.cpp

10.65 GfxDrvDXGIAdapterEnumerator Class Reference

Static Public Member Functions

- static GfxDrvDXGIAdapterList ∗ **EnumerateAdapters** (IDXGIFactory ∗factory)
- static void **DeleteAdapterList** (GfxDrvDXGIAdapterList ∗adapters)

The documentation for this class was generated from the following files:

- GfxDrvDXGIAdapterEnumerator.h
- GfxDrvDXGIAdapterEnumerator.cpp

10.66 GfxDrvDXGIErrorLogger Class Reference

Static Public Member Functions

- static void **LogError** (const char ∗headline, const HRESULT hResult)

Static Private Member Functions

- static const char ∗ **GetErrorString** (const HRESULT hResult)

The documentation for this class was generated from the following files:

- GfxDrvDXGIErrorLogger.h
- GfxDrvDXGIErrorLogger.cpp

10.67 GfxDrvDXGIMode Class Reference

Public Member Functions

- std::string **GetModeDescriptionString** ()
- unsigned int **GetId** ()
- unsigned int **GetWidth** ()
- unsigned int **GetHeight** ()
- unsigned int **GetRefreshRate** ()
- DXGI_MODE_SCALING **GetScaling** ()
- DXGI_MODE_SCANLINE_ORDER **GetScanlineOrder** ()
- DXGI_MODE_DESC ∗ **GetDXGIModeDescription** ()
- bool **CanUseMode** ()
- **GfxDrvDXGIMode** (DXGI_MODE_DESC ∗desc)

Static Public Member Functions

- static unsigned int **GetNewId** ()

Private Member Functions

- const char ∗ **GetScalingDescription** ()
- const char ∗ **GetScanlineOrderDescription** ()

Private Attributes

- unsigned int **_id**
- DXGI_MODE_DESC **_dxgi_mode_description**

Static Private Attributes

- static unsigned int **_next_id** = 1

The documentation for this class was generated from the following files:

- GfxDrvDXGIMode.h
- GfxDrvDXGIMode.cpp

10.68 GfxDrvDXGIModeEnumerator Class Reference

Static Public Member Functions

- static void **EnumerateModes** (IDXGIOutput ∗output, GfxDrvDXGIModeList &modes)
- static void **DeleteModeList** (GfxDrvDXGIModeList &modes)

The documentation for this class was generated from the following files:

- GfxDrvDXGIModeEnumerator.h
- GfxDrvDXGIModeEnumerator.cpp

10.69 GfxDrvDXGIOutput Class Reference

Public Member Functions

- const GfxDrvDXGIModeList & **GetModes** ()
- **GfxDrvDXGIOutput** (IDXGIOutput ∗output)

Private Member Functions

- const char ∗ **GetRotationDescription** (DXGI_MODE_ROTATION rotation)
- void **EnumerateModes** (IDXGIOutput ∗output)
- void **LogCapabilities** (IDXGIOutput ∗output)

Private Attributes

- char **_name** [255]
- GfxDrvDXGIModeList **_modes**

The documentation for this class was generated from the following files:

- GfxDrvDXGIOutput.h
- GfxDrvDXGIOutput.cpp

10.70 GfxDrvDXGIOutputEnumerator Class Reference

Static Public Member Functions

- static void **EnumerateOutputs** (IDXGIAdapter ∗adapter, GfxDrvDXGIOutputList &outputs)
- static void **DeleteOutputs** (GfxDrvDXGIOutputList &outputs)

The documentation for this class was generated from the following files:

- GfxDrvDXGIOutputEnumerator.h
- GfxDrvDXGIOutputEnumerator.cpp

10.71 graph_line Struct Reference

Data Fields

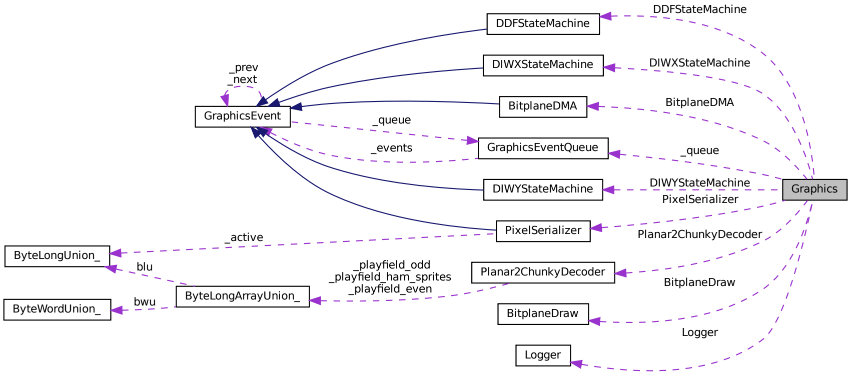
- graph_linetypes **linetype**
- UBY **line1** [1024]
- UBY **line2** [1024]
- ULO **colors** [64]
- ULO **DIW_first_draw**
- ULO **DIW_pixel_count**
- ULO **BG_pad_front**
- ULO **BG_pad_back**
- void ∗ **draw_line_routine**
- void ∗ **draw_line_BPL_res_routine**
- ULO **DDF_start**
- ULO **frames_left_until_BG_skip**
- ULO **sprite_ham_slot**
- ULO **bplcon2**
- bool **has_ham_sprites_online**

The documentation for this struct was generated from the following file:

- GRAPH.H

10.72 Graphics Class Reference

Collaboration diagram for Graphics:



Public Member Functions

- void **Commit** (ULO untilRasterY, ULO untilRasterX)
- void **EndOfFrame** (void)
- void **SoftReset** (void)
- void **HardReset** (void)
- void **EmulationStart** (void)
- void **EmulationStop** (void)
- void **Startup** (void)
- void **Shutdown** (void)

Data Fields

- DIWXStateMachine **DIWXStateMachine**
- DIWYStateMachine **DIWYStateMachine**
- DDFStateMachine **DDFStateMachine**
- BitplaneDMA **BitplaneDMA**
- PixelSerializer **PixelSerializer**
- Planar2ChunkyDecoder **Planar2ChunkyDecoder**
- BitplaneDraw **BitplaneDraw**
- Logger **Logger**

Private Member Functions

- void **InitializeEventQueue** (void)
- void **InitializeDIWXEvent** (void)
- void **InitializeDIWYEvent** (void)
- void **InitializeDDFEvent** (void)
- void **InitializeBitplaneDMAEvent** (void)
- void **InitializePixelSerializerEvent** (void)

Private Attributes

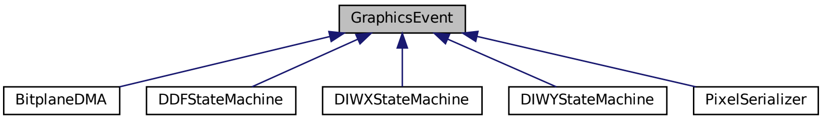
- GraphicsEventQueue **_queue**

The documentation for this class was generated from the following files:

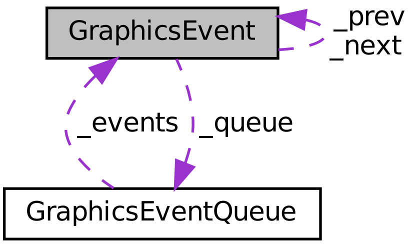
- Graphics.h
- Graphics.cpp

10.73 GraphicsEvent Class Reference

Inheritance diagram for GraphicsEvent:



Collaboration diagram for GraphicsEvent:



Public Member Functions

- ULO **MakeArriveTime** (ULO rasterY, ULO cylinder)
- virtual void **InitializeEvent** (GraphicsEventQueue ∗queue)=0
- virtual void **Handler** (ULO rasterY, ULO cylinder)=0

Data Fields

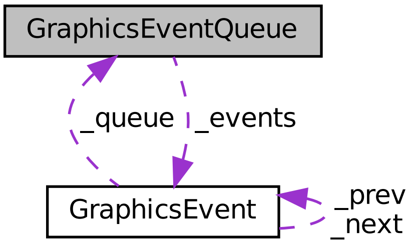
- GraphicsEventQueue ∗ **_queue**
- GraphicsEvent ∗ **_next**
- GraphicsEvent ∗ **_prev**
- ULO **_arriveTime**
- ULO **_priority**

The documentation for this class was generated from the following files:

- GraphicsEvent.h
- GraphicsEvent.cpp

10.74 GraphicsEventQueue Class Reference

Collaboration diagram for GraphicsEventQueue:



Public Member Functions

- void **Clear** (void)
- GraphicsEvent ∗ **Pop** (void)
- void **Insert** (GraphicsEvent ∗graphics_event)
- void **Remove** (GraphicsEvent ∗graphics_event)
- void **Run** (ULO untilTime)

Static Public Member Functions

- static ULO **GetCylindersPerLine** ()

Static Public Attributes

- static const ULO **GRAPHICS_ARRIVE_TIME_NONE** = 0xffffffff

Private Member Functions

- void **RunQueue** (ULO untilTime)

Private Attributes

- GraphicsEvent ∗ **_events**

The documentation for this class was generated from the following files:

- GraphicsEventQueue.h
- GraphicsEventQueue.cpp

10.75 gz_header_s Struct Reference

Data Fields

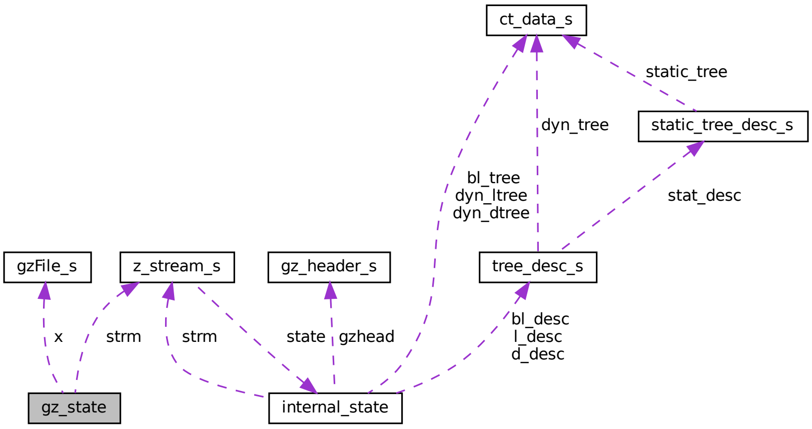
- int **text**
- uLong **time**
- int **xflags**
- int **os**
- Bytef ∗ **extra**
- uInt **extra_len**
- uInt **extra_max**
- Bytef ∗ **name**
- uInt **name_max**
- Bytef ∗ **comment**
- uInt **comm_max**
- int **hcrc**
- int **done**

The documentation for this struct was generated from the following file:

- zlib.h

10.76 gz_state Struct Reference

Collaboration diagram for gz_state:



Data Fields

- struct gzFile_s **x**
- int **mode**
- int **fd**
- char ∗ **path**
- unsigned **size**
- unsigned **want**
- unsigned char ∗ **in**
- unsigned char ∗ **out**
- int **direct**
- int **how**
- z_off64_t **start**
- int **eof**
- int **past**
- int **level**
- int **strategy**
- z_off64_t **skip**
- int **seek**
- int **err**
- char ∗ **msg**
- z_stream **strm**

The documentation for this struct was generated from the following file:

- gzguts.h

10.77 gzFile_s Struct Reference

Data Fields

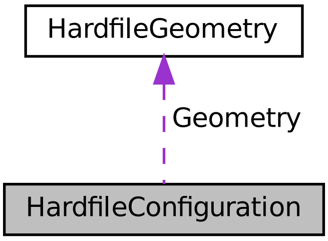
- unsigned **have**
- unsigned char ∗ **next**
- z_off64_t **pos**

The documentation for this struct was generated from the following file:

- zlib.h

10.78 HardfileConfiguration Struct Reference

Collaboration diagram for HardfileConfiguration:



Public Member Functions

- bool **operator==** (const HardfileConfiguration &other) const
- void **Clear** ()

Data Fields

- std::string **Filename**
- bool **Readonly**
- HardfileGeometry **Geometry**
- std::vector< HardfilePartition > **Partitions**

The documentation for this struct was generated from the following file:

- IHardfileHandler.h

10.79 hardfiledata Struct Reference

Data Fields

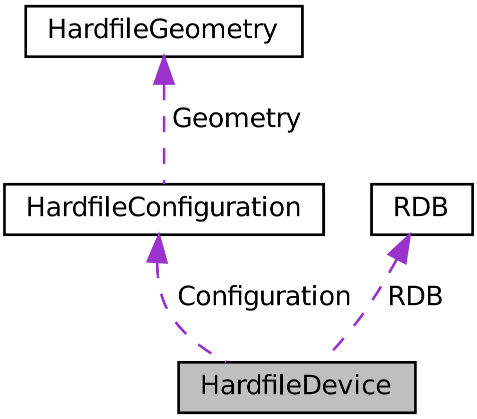
- unsigned long **size**
- int **nrcyls**
- int **secspertrack**
- int **surfaces**
- int **reservedblocks**
- int **blocksize**
- FILE ∗ **fd**

The documentation for this struct was generated from the following file:

- FILESYS.H

10.80 HardfileDevice Class Reference

Collaboration diagram for HardfileDevice:



Public Member Functions

- bool **CloseFile** ()
- void **DeleteRDB** ()
- bool **Clear** ()

Data Fields

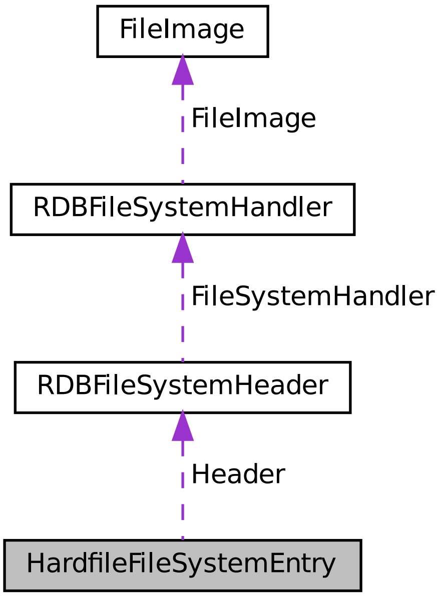
- fellow::api::module::HardfileConfiguration **Configuration**
- bool **Readonly**
- unsigned int **FileSize**
- unsigned int **GeometrySize**
- fhfile_status **Status**
- FILE ∗ **F**
- bool **HasRDB**
- rdb::RDB ∗ **RDB**

The documentation for this class was generated from the following file:

- HardfileStructs.h

10.81 HardfileFileSystemEntry Struct Reference

Collaboration diagram for HardfileFileSystemEntry:



Public Member Functions

- bool **IsOlderOrSameFileSystemVersion** (ULO DOSType, ULO version)
- bool **IsDOSType** (ULO DOSType)
- bool **IsOlderVersion** (ULO version)
- bool **IsOlderOrSameVersion** (ULO version)
- ULO **GetDOSType** ()
- ULO **GetVersion** ()
- void **CopyHunkToAddress** (ULO destinationAddress, ULO hunkIndex)
- **HardfileFileSystemEntry** (rdb::RDBFileSystemHeader ∗header, ULO segListAddress)

Data Fields

- rdb::RDBFileSystemHeader ∗ **Header**
- ULO **SegListAddress**

The documentation for this struct was generated from the following files:

- HardfileStructs.h
- HardfileStructs.cpp

10.82 HardfileGeometry Struct Reference

Public Member Functions

- void **Clear** ()
- bool **operator==** (const HardfileGeometry &other) const

Data Fields

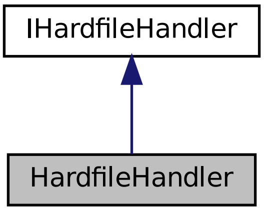
- unsigned int **LowCylinder**
- unsigned int **HighCylinder**
- unsigned int **BytesPerSector**
- unsigned int **SectorsPerTrack**
- unsigned int **Surfaces**
- unsigned int **Tracks**
- unsigned int **ReservedBlocks**

The documentation for this struct was generated from the following file:

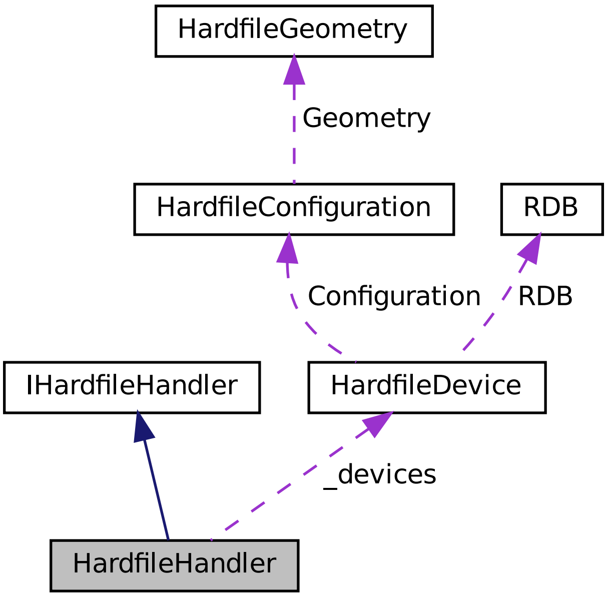
- IHardfileHandler.h

10.83 HardfileHandler Class Reference

Inheritance diagram for HardfileHandler:



Collaboration diagram for HardfileHandler:



Public Member Functions

- void **CardInit** () override
- void **CardMap** (ULO mapping) override
- UBY **ReadByte** (ULO address) override
- UWO **ReadWord** (ULO address) override
- ULO **ReadLong** (ULO address) override

- void **Do** (ULO data) override
- void **SetEnabled** (bool enabled) override
- bool **GetEnabled** () override
- void **Clear** () override
- bool **CompareHardfile** (const fellow::api::module::HardfileConfiguration &configuration, unsigned int index) override
- void **SetHardfile** (const fellow::api::module::HardfileConfiguration &configuration, unsigned int index) override
- bool **RemoveHardfile** (unsigned int index) override
- unsigned int **GetMaxHardfileCount** () override
- void **SetUnitNoStartNumber** (unsigned int unitNoStartNumber) override
- bool **Create** (const fellow::api::module::HardfileConfiguration &configuration, ULO size) override
- bool **HasRDB** (const std::string &filename) override
- fellow::api::module::HardfileConfiguration **GetConfigurationFromRDBGeometry** (const std::string &filename) override
- void **EmulationStart** () override
- void **EmulationStop** () override
- void **HardReset** () override
- void **Startup** () override
- void **Shutdown** () override

Private Member Functions

- bool **HasZeroDevices** ()
- void **CreateMountList** ()
- std::string **MakeDeviceName** ()
- std::string **MakeDeviceName** (const std::string &preferredName)
- bool **PreferredNameExists** (const std::string &preferredName)
- bool **FindOlderOrSameFileSystemVersion** (ULO DOSType, ULO version, unsigned int &olderOrSame↩FileSystemIndex)
- HardfileFileSystemEntry ∗ **GetFileSystemForDOSType** (ULO DOSType)
- void **AddFileSystemsFromRdb** (HardfileDevice &device)
- void **AddFileSystemsFromRdb** ()
- void **EraseOlderOrSameFileSystemVersion** (ULO DOSType, ULO version)
- void **SetHardfileConfigurationFromRDB** (fellow::api::module::HardfileConfiguration &config, rdb::RDB ∗rdb, bool readonly)
- void **InitializeHardfile** (unsigned int index)
- void **Ignore** (ULO index)
- BYT **Read** (ULO index)
- BYT **Write** (ULO index)
- void **GetNumberOfTracks** (ULO index)
- void **GetDriveType** (ULO index)
- void **WriteProt** (ULO index)
- void **DoDiag** ()
- void **DoOpen** ()
- void **DoClose** ()
- void **DoExpunge** ()
- void **DoNULL** ()
- void **DoBeginIO** ()
- void **DoAbortIO** ()
- ULO **DoGetRDBFileSystemCount** ()
- ULO **DoGetRDBFileSystemHunkCount** (ULO fileSystemIndex)
- ULO **DoGetRDBFileSystemHunkSize** (ULO fileSystemIndex, ULO hunkIndex)
- void **DoCopyRDBFileSystemHunk** (ULO destinationAddress, ULO fileSystemIndex, ULO hunkIndex)

- void **DoRelocateFileSystem** (ULO fileSystemIndex)
- void **DoInitializeRDBFileSystemEntry** (ULO fileSystemEntry, ULO fileSystemIndex)
- void **DoPatchDOSDeviceNode** (ULO node, ULO packet)
- void **DoUnknownOperation** (ULO operation)
- std::string **LogGetStringFromMemory** (ULO address)
- void **DoLogAvailableFileSystems** (ULO fileSystemResource)
- void **DoLogAvailableResources** ()
- void **DoLogAllocMemResult** (ULO result)
- void **DoLogOpenResourceResult** (ULO result)
- void **DoRemoveRDBFileSystemsAlreadySupportedBySystem** (ULO filesystemResource)
- void **MakeDOSDevPacketForPlainHardfile** (const HardfileMountListEntry &mountListEntry, ULO device↩ NameAddress)
- void **MakeDOSDevPacketForRDBPartition** (const HardfileMountListEntry &mountListEntry, ULO device↩ NameAddress)

Private Attributes

- HardfileDevice **_devices** [FHFILE_MAX_DEVICES]
- std::vector< std::unique_ptr< HardfileFileSystemEntry > > **_fileSystems**
- std::vector< std::unique_ptr< HardfileMountListEntry > > **_mountList**
- ULO **_romstart** = 0
- ULO **_bootcode** = 0
- ULO **_configdev** = 0
- ULO **_fsname** = 0
- UBY **_rom** [65536]
- bool **_enabled** = false
- unsigned int **_unitNoStartNumber**

The documentation for this class was generated from the following files:

- HardfileHandler.h
- HardfileHandler.cpp

10.84 HardfileMountListEntry Struct Reference

Public Member Functions

- **HardfileMountListEntry** (unsigned int deviceIndex, int partitionIndex, const std::string &name)

Data Fields

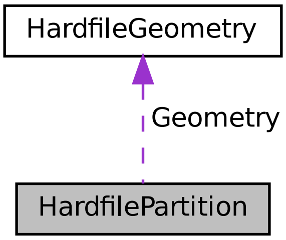
- unsigned int **DeviceIndex**
- int **PartitionIndex**
- std::string **Name**
- ULO **NameAddress**

The documentation for this struct was generated from the following file:

- HardfileStructs.h

10.85 HardfilePartition Struct Reference

Collaboration diagram for HardfilePartition:



Data Fields

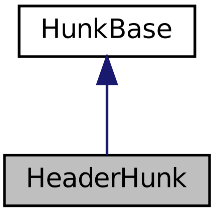
- std::string **PreferredName**
- HardfileGeometry **Geometry**

The documentation for this struct was generated from the following file:

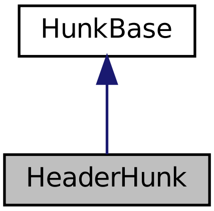
- IHardfileHandler.h

10.86 HeaderHunk Class Reference

Inheritance diagram for HeaderHunk:



Collaboration diagram for HeaderHunk:



Public Member Functions

- ULO **GetID** () override
- ULO **GetHunkSizeCount** ()
- const HunkSize & **GetHunkSize** (ULO index)
- ULO **GetResidentLibraryCount** ()
- const std::string & **GetResidentLibrary** (ULO index)
- ULO **GetFirstLoadHunk** ()
- ULO **GetLastLoadHunk** ()
- void **Parse** (RawDataReader &rawDataReader) override

Private Attributes

- std::vector< std::string > **_residentLibraries**
- std::vector< HunkSize > **_hunkSizes**
- ULO **_firstLoadHunk**
- ULO **_lastLoadHunk**

Static Private Attributes

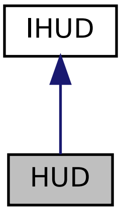
- static const ULO **ID** = HeaderHunkID

The documentation for this class was generated from the following files:

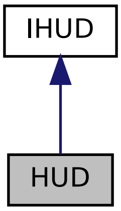
- HeaderHunk.h
- HeaderHunk.cpp

10.87 HUD Class Reference

Inheritance diagram for HUD:



Collaboration diagram for HUD:



Public Member Functions

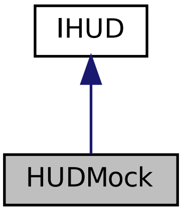
- void **SetFloppyLED** (int driveIndex, bool active, bool write) override
- void **SetHarddiskLED** (int deviceIndex, bool active, bool write) override

The documentation for this class was generated from the following files:

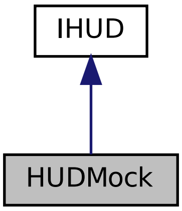
- HUD.h
- HUD.cpp

10.88 HUDMock Class Reference

Inheritance diagram for HUDMock:



Collaboration diagram for HUDMock:



Public Member Functions

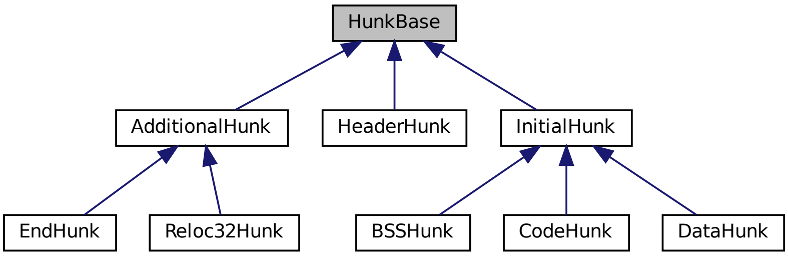
- void **SetFloppyLED** (int driveIndex, bool active, bool write) override
- void **SetHarddiskLED** (int deviceIndex, bool active, bool write) override

The documentation for this class was generated from the following files:

- HUDMock.h
- HUDMock.cpp

10.89 HunkBase Class Reference

Inheritance diagram for HunkBase:



Public Member Functions

- virtual ULO **GetID** ()=0
- virtual void **Parse** (RawDataReader &rawReader)=0

The documentation for this class was generated from the following file:

- HunkBase.h

10.90 HunkFactory Class Reference

Static Public Member Functions

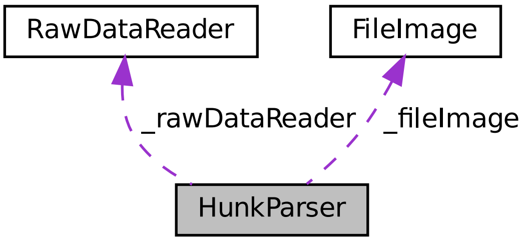
- static InitialHunk ∗ **CreateInitialHunk** (ULO type, ULO allocateSizeInLongwords)
- static AdditionalHunk ∗ **CreateAdditionalHunk** (ULO type, ULO sourceHunkIndex)

The documentation for this class was generated from the following files:

- HunkFactory.h
- HunkFactory.cpp

10.91 HunkParser Class Reference

Collaboration diagram for HunkParser:



Public Member Functions

- bool **Parse** ()
- **HunkParser** (UBY ∗rawData, ULO rawDataLength, FileImage &fileImage)

Private Member Functions

- HeaderHunk ∗ **ParseHeader** ()
- InitialHunk ∗ **ParseNextInitialHunk** (ULO allocateSizeInLongwords)
- AdditionalHunk ∗ **ParseNextAdditionalHunk** (ULO sourceHunkIndex)

Private Attributes

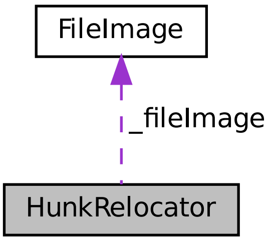
- RawDataReader **_rawDataReader**
- FileImage & **_fileImage**

The documentation for this class was generated from the following files:

- HunkParser.h
- HunkParser.cpp

10.92 HunkRelocator Class Reference

Collaboration diagram for HunkRelocator:



Public Member Functions

- void **RelocateHunks** ()
- **HunkRelocator** (FileImage &fileImage)

Private Member Functions

- void **ProcessReloc32OffsetTable** (Reloc32OffsetTable ∗offsetTable, ULO hunkBaseAddress)
- void **ProcessReloc32Hunk** (Reloc32Hunk ∗reloc32Hunk, ULO hunkBaseAddress)
- void **RelocateHunk** (ULO hunkIndex)

Private Attributes

- FileImage & **_fileImage**

The documentation for this class was generated from the following files:

- HunkRelocator.h
- HunkRelocator.cpp

10.93 HunkSize Struct Reference

Public Member Functions

- const STR ∗ **GetMemoryFlagsToString** ()
- **HunkSize** (ULO sizeInLongwords, ULO memoryFlags, ULO additionalFlags)

Data Fields

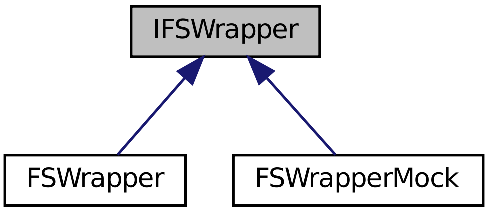
- ULO **SizeInLongwords**
- ULO **MemoryFlags**
- ULO **AdditionalFlags**

The documentation for this struct was generated from the following files:

- HunkSize.h
- HunkSize.cpp

10.94 IFSWrapper Class Reference

Inheritance diagram for IFSWrapper:



Public Member Functions

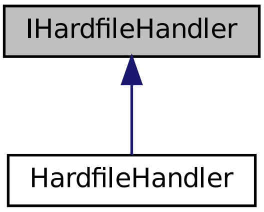
- virtual fs_wrapper_point ∗ **MakePoint** (const STR ∗point)=0

The documentation for this class was generated from the following file:

- IFSWrapper.h

10.95 IHardfileHandler Class Reference

Inheritance diagram for IHardfileHandler:



Public Member Functions

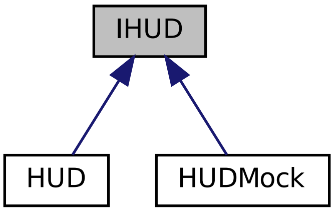
- virtual void **CardInit** ()=0
- virtual void **CardMap** (ULO mapping)=0
- virtual UBY **ReadByte** (ULO address)=0
- virtual UWO **ReadWord** (ULO address)=0
- virtual ULO **ReadLong** (ULO address)=0
- virtual void **Do** (ULO data)=0
- virtual void **SetEnabled** (bool enabled)=0
- virtual bool **GetEnabled** ()=0
- virtual void **Clear** ()=0
- virtual bool **CompareHardfile** (const HardfileConfiguration &hardfile, unsigned int index)=0
- virtual void **SetHardfile** (const HardfileConfiguration &hardfile, unsigned int index)=0
- virtual bool **RemoveHardfile** (unsigned int index)=0
- virtual unsigned int **GetMaxHardfileCount** ()=0
- virtual void **SetUnitNoStartNumber** (unsigned int unitNoStartNumber)=0
- virtual bool **Create** (const HardfileConfiguration &configuration, ULO size)=0
- virtual bool **HasRDB** (const std::string &filename)=0
- virtual HardfileConfiguration **GetConfigurationFromRDBGeometry** (const std::string &filename)=0
- virtual void **EmulationStart** ()=0
- virtual void **EmulationStop** ()=0
- virtual void **HardReset** ()=0
- virtual void **Startup** ()=0
- virtual void **Shutdown** ()=0

The documentation for this class was generated from the following file:

- IHardfileHandler.h

10.96 IHUD Class Reference

Inheritance diagram for IHUD:



Public Member Functions

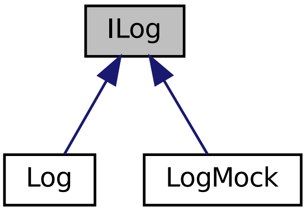
- virtual void **SetFloppyLED** (int driveIndex, bool active, bool write)=0
- virtual void **SetHarddiskLED** (int deviceIndex, bool active, bool write)=0

The documentation for this class was generated from the following file:

- IHUD.h

10.97 ILog Class Reference

Inheritance diagram for ILog:



Public Member Functions

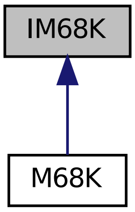
- virtual void **AddLogDebug** (const char ∗format,...)=0
- virtual void **AddLog** (const char ∗,...)=0
- virtual void **AddLog2** (STR ∗msg)=0

The documentation for this class was generated from the following file:

- ILog.h

10.98 IM68K Class Reference

Inheritance diagram for IM68K:



Public Member Functions

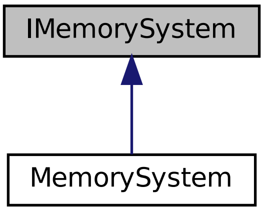
- virtual void **SetDReg** (ULO registerNumber, ULO value)=0
- virtual ULO **GetDReg** (ULO registerNumber)=0
- virtual ULO **GetAReg** (ULO registerNumber)=0

The documentation for this class was generated from the following file:

- IM68K.h

10.99 IMemorySystem Class Reference

Inheritance diagram for IMemorySystem:



Public Member Functions

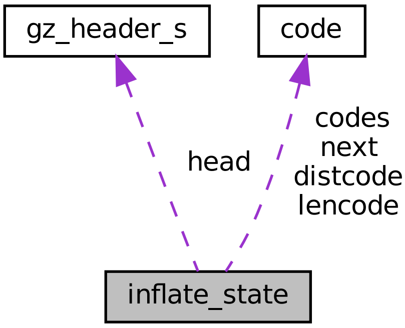
- virtual UBY **ReadByte** (ULO address)=0
- virtual UWO **ReadWord** (ULO address)=0
- virtual ULO **ReadLong** (ULO address)=0
- virtual void **WriteByte** (UBY data, ULO address)=0
- virtual void **WriteWord** (UWO data, ULO address)=0
- virtual void **WriteLong** (ULO data, ULO address)=0
- virtual void **DmemSetByte** (UBY data)=0
- virtual void **DmemSetWord** (UWO data)=0
- virtual void **DmemSetLong** (ULO data)=0
- virtual void **DmemSetLongNoCounter** (ULO data, ULO offset)=0
- virtual void **DmemSetString** (const STR ∗data)=0
- virtual void **DmemSetCounter** (ULO val)=0
- virtual ULO **DmemGetCounter** ()=0
- virtual void **DmemClear** ()=0
- virtual void **EmemClear** ()=0
- virtual void **EmemSet** (ULO index, ULO data)=0
- virtual void **EmemCardAdd** (EmemCardInitFunc cardinit, EmemCardMapFunc cardmap)=0
- virtual void **EmemMirror** (ULO emem_offset, UBY ∗src, ULO size)=0
- virtual void **BankSet** (ReadByteFunc rb, ReadWordFunc rw, ReadLongFunc rl, WriteByteFunc wb, Write←WordFunc ww, WriteLongFunc wl, UBY ∗basep, ULO bank, ULO basebank, BOOLE pointer_can_write)=0
- virtual UBY ∗ **AddressToPtr** (ULO address)=0
- virtual ULO **GetKickImageVersion** ()=0

The documentation for this class was generated from the following file:

- IMemorySystem.h

10.100 inflate_state Struct Reference

Collaboration diagram for inflate_state:



Data Fields

- inflate_mode **mode**
- int **last**
- int **wrap**
- int **havedict**
- int **flags**
- unsigned **dmax**
- unsigned long **check**
- unsigned long **total**
- gz_headerp **head**
- unsigned **wbits**
- unsigned **wsize**
- unsigned **whave**
- unsigned **wnext**
- unsigned char FAR ∗ **window**
- unsigned long **hold**
- unsigned **bits**
- unsigned **length**
- unsigned **offset**
- unsigned **extra**
- code const FAR ∗ **lencode**
- code const FAR ∗ **distcode**
- unsigned **lenbits**
- unsigned **distbits**
- unsigned **ncode**
- unsigned **nlen**
- unsigned **ndist**
- unsigned **have**
- code FAR ∗ **next**
- unsigned short **lens** [320]

- unsigned short **work** [288]
- code **codes** [ENOUGH]
- int **sane**
- int **back**
- unsigned **was**

The documentation for this struct was generated from the following file:

- inflate.h

10.101 ini Struct Reference

Data Fields

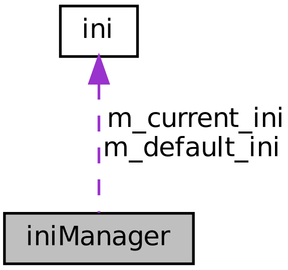
- STR **m_description** [256]
- STR **m_current_configuration** [CFG_FILENAME_LENGTH]
- int **m_mainwindowxposition**
- int **m_mainwindowyposition**
- int **m_emulationwindowxposition**
- int **m_emulationwindowyposition**
- STR **m_configuration_history** [4][CFG_FILENAME_LENGTH]
- STR **m_lastusedkeydir** [CFG_FILENAME_LENGTH]
- STR **m_lastusedkickimagedir** [CFG_FILENAME_LENGTH]
- STR **m_lastusedconfigurationdir** [CFG_FILENAME_LENGTH]
- ULO **m_lastusedconfigurationtab**
- STR **m_lastusedglobaldiskdir** [CFG_FILENAME_LENGTH]
- STR **m_lastusedhdfdir** [CFG_FILENAME_LENGTH]
- STR **m_lastusedmoddir** [CFG_FILENAME_LENGTH]
- STR **m_lastusedstatefiledir** [CFG_FILENAME_LENGTH]
- STR **m_lastusedpresetromdir** [CFG_FILENAME_LENGTH]

The documentation for this struct was generated from the following file:

- Ini.h

10.102 iniManager Struct Reference

Collaboration diagram for iniManager:



Data Fields

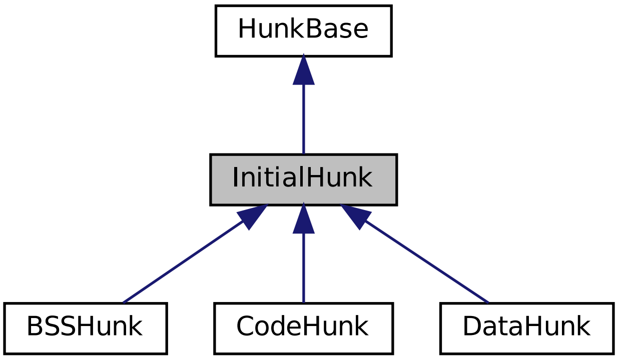
- ini ∗ **m_current_ini**
- ini ∗ **m_default_ini**

The documentation for this struct was generated from the following file:

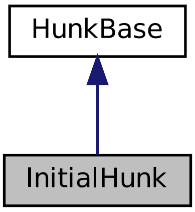
- Ini.h

10.103 InitialHunk Class Reference

Inheritance diagram for InitialHunk:



Collaboration diagram for InitialHunk:



Public Member Functions

- void **Parse** (RawDataReader &rawReader) override=0
- ULO **GetAllocateSizeInLongwords** ()
- ULO **GetAllocateSizeInBytes** ()
- ULO **GetContentSizeInLongwords** ()
- ULO **GetContentSizeInBytes** ()
- UBY ∗ **GetContent** ()
- void **SetVMAddress** (ULO vmAddress)
- ULO **GetVMAddress** ()
- **InitialHunk** (ULO allocateSizeInLongwords)

Protected Attributes

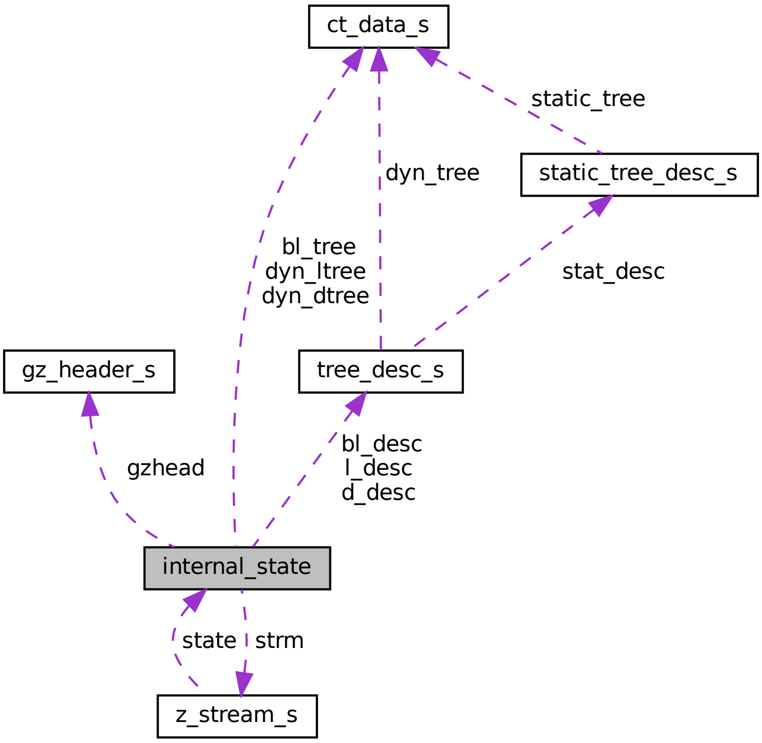
- ULO **_allocateSizeInLongwords**
- ULO **_contentSizeInLongwords**
- ULO **_vmAddress**
- std::unique_ptr< UBY > **_rawData**

The documentation for this class was generated from the following files:

- InitialHunk.h
- InitialHunk.cpp

10.104 internal_state Struct Reference

Collaboration diagram for internal_state:



Data Fields

- int **dummy**
- z_streamp **strm**
- int **status**
- Bytef ∗ **pending_buf**
- ulg **pending_buf_size**
- Bytef ∗ **pending_out**
- uInt **pending**
- int **wrap**
- gz_headerp **gzhead**
- uInt **gzindex**
- Byte **method**
- int **last_flush**
- uInt **w_size**
- uInt **w_bits**
- uInt **w_mask**
- Bytef ∗ **window**
- ulg **window_size**
- Posf ∗ **prev**
- Posf ∗ **head**
- uInt **ins_h**
- uInt **hash_size**
- uInt **hash_bits**
- uInt **hash_mask**
- uInt **hash_shift**
- long **block_start**
- uInt **match_length**
- IPos **prev_match**
- int **match_available**
- uInt **strstart**
- uInt **match_start**
- uInt **lookahead**
- uInt **prev_length**
- uInt **max_chain_length**
- uInt **max_lazy_match**
- int **level**
- int **strategy**
- uInt **good_match**
- int **nice_match**
- struct ct_data_s **dyn_ltree** [HEAP_SIZE]
- struct ct_data_s **dyn_dtree** [2 ∗D_CODES+1]
- struct ct_data_s **bl_tree** [2 ∗BL_CODES+1]
- struct tree_desc_s **l_desc**
- struct tree_desc_s **d_desc**
- struct tree_desc_s **bl_desc**
- ush **bl_count** [MAX_BITS+1]
- int **heap** [2 ∗L_CODES+1]
- int **heap_len**
- int **heap_max**
- uch **depth** [2 ∗L_CODES+1]
- uchf ∗ **l_buf**
- uInt **lit_bufsize**
- uInt **last_lit**
- ushf ∗ **d_buf**

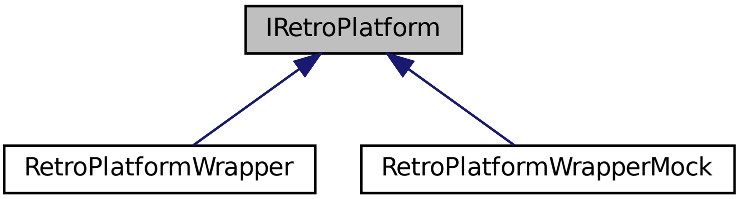
- ulg **opt_len**
- ulg **static_len**
- uInt **matches**
- uInt **insert**
- ush **bi_buf**
- int **bi_valid**
- ulg **high_water**

The documentation for this struct was generated from the following files:

- zutil.c
- deflate.h
- zlib.h

10.105 IRetroPlatform Class Reference

Inheritance diagram for IRetroPlatform:



Public Member Functions

- virtual bool **SendHardDriveContent** (const ULO lHardDriveNo, const STR ∗szImageName, const bool b↩ WriteProtected)=0
- virtual bool **PostHardDriveLED** (const ULO lHardDriveNo, const bool bActive, const bool bWriteActivity)=0

The documentation for this class was generated from the following file:

- IRetroPlatform.h

10.106 kbd_buffer_type Struct Reference

Data Fields

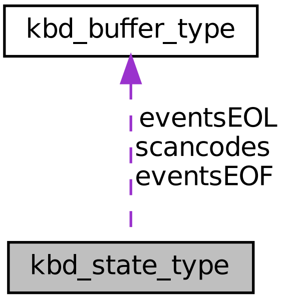
- UBY **buffer** [KBDBUFFERLENGTH]
- ULO **inpos**
- ULO **outpos**

The documentation for this struct was generated from the following file:

- KBD.H

10.107 kbd_state_type Struct Reference

Collaboration diagram for kbd_state_type:



Data Fields

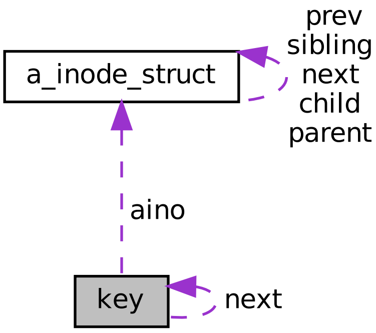
- kbd_buffer_type **scancodes**
- kbd_buffer_type **eventsEOL**
- kbd_buffer_type **eventsEOF**

The documentation for this struct was generated from the following file:

- KBD.H

10.108 key Struct Reference

Collaboration diagram for key:



Data Fields

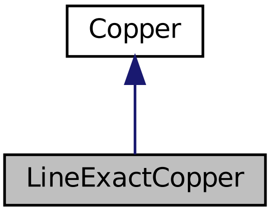
- struct key ∗ **next**
- a_inode ∗ **aino**
- uae_u32 **uniq**
- int **fd**
- off_t **file_pos**

The documentation for this struct was generated from the following file:

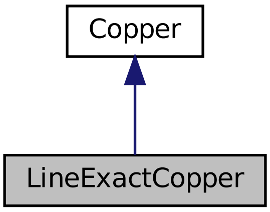
- FILESYS.C

10.109 LineExactCopper Class Reference

Inheritance diagram for LineExactCopper:



Collaboration diagram for LineExactCopper:



Public Member Functions

- virtual void **NotifyDMAEnableChanged** (bool new_dma_enable_state)
- virtual void **NotifyCop1lcChanged** ()
- virtual void **Load** (ULO new_copper_pc)
- virtual void **EventHandler** ()
- virtual void **EndOfFrame** ()
- virtual void **HardReset** ()
- virtual void **EmulationStart** ()
- virtual void **EmulationStop** ()

Private Member Functions

- void **YTableInit** ()
- ULO **GetCheckedWaitCycle** (ULO waitCycle)
- void **RemoveEvent** ()
- void **InsertEvent** (ULO cycle)

Private Attributes

- ULO **ytable** [512]

Static Private Attributes

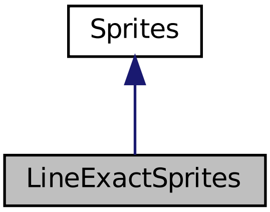
- static ULO **cycletable** [16] = { 4, 4, 4, 4, 4, 5, 6, 4, 4, 4, 4, 8, 16, 4, 4, 4 }

The documentation for this class was generated from the following files:

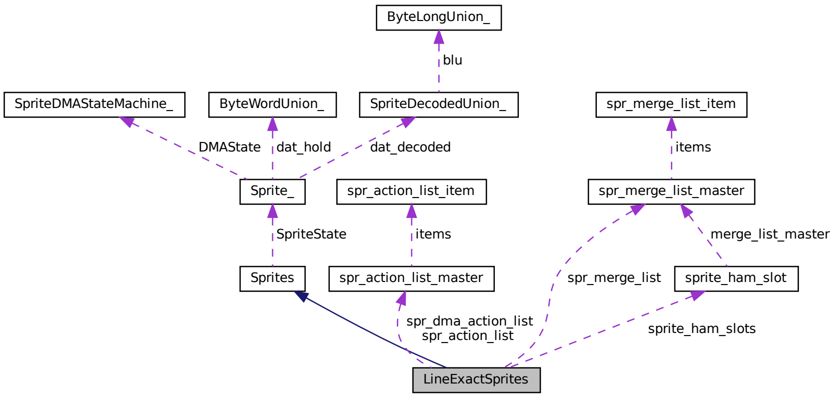
- LineExactCopper.h
- LineExactCopper.cpp

10.110 LineExactSprites Class Reference

Inheritance diagram for LineExactSprites:



Collaboration diagram for LineExactSprites:



Data Structures

- struct sprite_ham_slot

Public Member Functions

- bool **HasSpritesOnLine** ()
- void **DMASpriteHandler** ()
- void **ProcessActionList** ()
- void **Merge** (graph_line ∗current_graph_line)
- void **MergeHAM2x1x16** (ULO ∗frameptr, graph_line ∗linedescription)
- void **MergeHAM2x2x16** (ULO ∗frameptr, graph_line ∗linedescription, ULO nextlineoffset)
- void **MergeHAM4x2x16** (ULL ∗frameptr, graph_line ∗linedescription, ULO nextlineoffset)
- void **MergeHAM4x4x16** (ULL ∗frameptr, graph_line ∗linedescription, ULO nextlineoffset, ULO nextlineoffset2, ULO nextlineoffset3)
- void **MergeHAM2x1x24** (UBY ∗frameptr, graph_line ∗linedescription)
- void **MergeHAM2x2x24** (UBY ∗frameptr, graph_line ∗linedescription, ULO nextlineoffset)
- void **MergeHAM4x2x24** (UBY ∗frameptr, graph_line ∗linedescription, ULO nextlineoffset)
- void **MergeHAM4x4x24** (UBY ∗frameptr, graph_line ∗linedescription, ULO nextlineoffset, ULO nextlineoffset2, ULO nextlineoffset3)
- void **MergeHAM2x1x32** (ULL ∗frameptr, graph_line ∗linedescription)
- void **MergeHAM2x2x32** (ULL ∗frameptr, graph_line ∗linedescription, ULO nextlineoffset)
- void **MergeHAM4x2x32** (ULL ∗frameptr, graph_line ∗linedescription, ULO nextlineoffset)
- void **MergeHAM4x4x32** (ULL ∗frameptr, graph_line ∗linedescription, ULO nextlineoffset, ULO nextlineoffset2, ULO nextlineoffset3)
- virtual void **NotifySprpthChanged** (UWO data, unsigned int sprite_number)
- virtual void **NotifySprptlChanged** (UWO data, unsigned int sprite_number)
- virtual void **NotifySprposChanged** (UWO data, unsigned int sprite_number)
- virtual void **NotifySprctlChanged** (UWO data, unsigned int sprite_number)
- virtual void **NotifySprdataChanged** (UWO data, unsigned int sprite_number)
- virtual void **NotifySprdatbChanged** (UWO data, unsigned int sprite_number)
- virtual void **HardReset** ()
- virtual void **EndOfLine** (ULO rasterY)
- virtual void **EndOfFrame** ()
- virtual void **EmulationStart** ()
- virtual void **EmulationStop** ()

Private Member Functions

- void **aspr0pth** (UWO data, ULO address)
- void **aspr0ptl** (UWO data, ULO address)
- void **aspr1pth** (UWO data, ULO address)
- void **aspr1ptl** (UWO data, ULO address)
- void **aspr2pth** (UWO data, ULO address)
- void **aspr2ptl** (UWO data, ULO address)
- void **aspr3pth** (UWO data, ULO address)
- void **aspr3ptl** (UWO data, ULO address)
- void **aspr4pth** (UWO data, ULO address)
- void **aspr4ptl** (UWO data, ULO address)
- void **aspr5pth** (UWO data, ULO address)
- void **aspr5ptl** (UWO data, ULO address)
- void **aspr6pth** (UWO data, ULO address)
- void **aspr6ptl** (UWO data, ULO address)
- void **aspr7pth** (UWO data, ULO address)
- void **aspr7ptl** (UWO data, ULO address)
- void **asprxpos** (UWO data, ULO address)
- void **asprxctl** (UWO data, ULO address)
- void **asprxdata** (UWO data, ULO address)
- void **asprxdatb** (UWO data, ULO address)
- spr_action_list_item ∗ **ActionListAddLast** (spr_action_list_master ∗l)
- ULO **ActionListCount** (spr_action_list_master ∗l)
- spr_action_list_item ∗ **ActionListGet** (spr_action_list_master ∗l, ULO i)
- void **ActionListClear** (spr_action_list_master ∗l)
- spr_action_list_item ∗ **ActionListAddSorted** (spr_action_list_master ∗l, ULO raster_x, ULO raster_y)
- spr_merge_list_item ∗ **MergeListAddLast** (spr_merge_list_master ∗l)
- ULO **MergeListCount** (spr_merge_list_master ∗l)
- spr_merge_list_item ∗ **MergeListGet** (spr_merge_list_master ∗l, ULO i)
- void **MergeListClear** (spr_merge_list_master ∗l)
- void **MergeHAM** (graph_line ∗linedescription)
- void **BuildItem** (spr_action_list_item ∗∗item)
- void **Log** ()
- void **ClearState** ()
- void **LogActiveSprites** ()
- void **Decode4Sprite** (ULO sprite_number)
- void **Decode16Sprite** (ULO sprite_number)
- void **SetDebugging** ()
- void **MergeDualLoresPF2loopinfront2** (graph_line ∗current_graph_line, ULO sprnr)
- void **MergeDualLoresPF1loopinfront2** (graph_line ∗current_graph_line, ULO sprnr)
- void **MergeDualLoresPF1loopbehind2** (graph_line ∗current_graph_line, ULO sprnr)
- void **MergeDualLoresPF2loopbehind2** (graph_line ∗current_graph_line, ULO sprnr)
- void **MergeDualLoresPlayfield** (graph_line ∗current_graph_line)
- void **MergeDualHiresPF2loopinfront2** (graph_line ∗current_graph_line, ULO sprnr)
- void **MergeDualHiresPF1loopinfront2** (graph_line ∗current_graph_line, ULO sprnr)
- void **MergeDualHiresPF1loopbehind2** (graph_line ∗current_graph_line, ULO sprnr)
- void **MergeDualHiresPF2loopbehind2** (graph_line ∗current_graph_line, ULO sprnr)
- void **MergeDualHiresPlayfield** (graph_line ∗current_graph_line)
- void **MergeHires** (graph_line ∗current_graph_line)
- void **MergeLores** (graph_line ∗current_graph_line)
- void **ProcessActionListNOP** ()
- void **ProcessDMAActionListNOP** ()

Private Attributes

- ULO **sprite_to_block**
- BOOLE **output_sprite_log**
- BOOLE **output_action_sprite_log**
- ULO **sprpt_debug** [8]
- ULO **sprx** [8]
- ULO **sprx_debug** [8]
- ULO **spry** [8]
- ULO **spry_debug** [8]
- ULO **sprly** [8]
- ULO **sprly_debug** [8]
- ULO **spratt** [8]
- UWO **sprdat** [8][2]
- BOOLE **spr_arm_data** [8]
- BOOLE **spr_arm_comparator** [8]
- spr_action_list_master **spr_action_list** [8]
- spr_action_list_master **spr_dma_action_list** [8]
- spr_merge_list_master **spr_merge_list** [8]
- STR **buffer** [128]
- ULO **sprite_state** [8]
- ULO **sprite_state_old** [8]
- ULO **sprite_16col** [8]
- ULO **sprite_online** [8]
- bool **sprites_online**
- UBY **sprite** [8][16]
- sprite_ham_slot **sprite_ham_slots** [313]
- ULO **sprite_ham_slot_next**
- ULO **sprite_write_buffer** [128][2]
- ULO **sprite_write_next**
- ULO **sprite_write_real**

Static Private Attributes

- static spr_register_func **sprxptl_functions** [8]
- static spr_register_func **sprxpth_functions** [8]

10.110.1 Field Documentation

10.110.1.1 sprxpth_functions

```
spr_register_func sprxpth_functions  [static], [private]
```

Initial value:

```
=
{
  &LineExactSprites::aspr0pth,
  &LineExactSprites::aspr1pth,
  &LineExactSprites::aspr2pth,
  &LineExactSprites::aspr3pth,
  &LineExactSprites::aspr4pth,
  &LineExactSprites::aspr5pth,
  &LineExactSprites::aspr6pth,
  &LineExactSprites::aspr7pth
}
```

10.110.1.2 sprxptl_functions

```
spr_register_func sprxptl_functions  [static], [private]
```

Initial value:

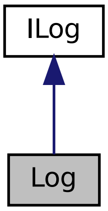
```
=
{
  &LineExactSprites::aspr0ptl,
  &LineExactSprites::aspr1ptl,
  &LineExactSprites::aspr2ptl,
  &LineExactSprites::aspr3ptl,
  &LineExactSprites::aspr4ptl,
  &LineExactSprites::aspr5ptl,
  &LineExactSprites::aspr6ptl,
  &LineExactSprites::aspr7ptl
}
```

The documentation for this class was generated from the following files:

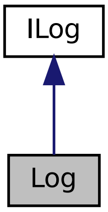
- LineExactSprites.h
- LineExactSprites.cpp

10.111 Log Class Reference

Inheritance diagram for Log:



Collaboration diagram for Log:



Public Member Functions

- void **AddLogDebug** (const char ∗format,...) override
- void **AddLog** (const char ∗,...) override
- void **AddLog2** (STR ∗msg) override

Private Member Functions

- STR ∗ **LogTime** (STR ∗buffer)

Private Attributes

- bool **_new_line**
- bool **_first_time**
- bool **_enabled**
- unsigned int **_level**
- std::string **_logfilename**

Static Private Attributes

- static const unsigned int **LogLevelError** = 0
- static const unsigned int **LogLevelInformation** = 1
- static const unsigned int **LogLevelDebug** = 2

The documentation for this class was generated from the following files:

- Log.h
- Log.cpp

10.112 Logger Class Reference

Public Member Functions

- void **Log** (ULO line, ULO cylinder, STR ∗message)
- bool **IsLogEnabled** (void)
- void **Shutdown** (void)

Private Attributes

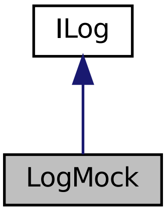
- bool **_enableLog**
- FILE ∗ **_logfile**

The documentation for this class was generated from the following files:

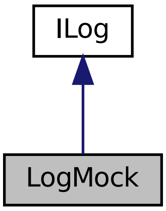
- Logger.h
- Logger.cpp

10.113 LogMock Class Reference

Inheritance diagram for LogMock:



Collaboration diagram for LogMock:



Public Member Functions

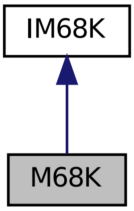
- void **AddLogDebug** (const char ∗,...) override
- void **AddLog** (const char ∗,...) override
- void **AddLog2** (STR ∗msg) override

The documentation for this class was generated from the following files:

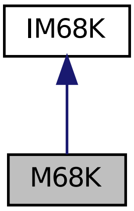
- LogMock.h
- LogMock.cpp

10.114 M68K Class Reference

Inheritance diagram for M68K:



Collaboration diagram for M68K:



Public Member Functions

- void **SetDReg** (ULO registerNumber, ULO value) override
- ULO **GetDReg** (ULO registerNumber) override
- ULO **GetAReg** (ULO registerNumber) override

The documentation for this class was generated from the following files:

- M68K.h
- M68K.cpp

10.115 m68k_cpu Class Reference

Public Member Functions

- uint32 **get_pc** () const
- void **set_pc** (uint32 pc)
- uint32 **get_ccr** () const
- void **set_ccr** (uint32 ccr)
- uint32 **get_dreg** (int r) const
- void **set_dreg** (int r, uint32 v)
- uint32 **get_areg** (int r) const
- void **set_areg** (int r, uint32 v)
- void **reset** (void)
- void **reset_jit** (void)
- void **execute** (uint32 pc)

Private Attributes

- void ∗ **opaque**

The documentation for this class was generated from the following files:

- m68k-tester.h
- m68k-tester-fellow.cpp

10.116 m68k_cpu_state_t Struct Reference

Data Fields

- uint32 **ccr**
- uint32 **dregs** [8]
- uint32 **aregs** [8]
- uint8 **use_dregs**
- uint8 **use_aregs**

The documentation for this struct was generated from the following file:

- m68k-tester.cpp

10.117 m68k_instruction_t Struct Reference

Data Fields

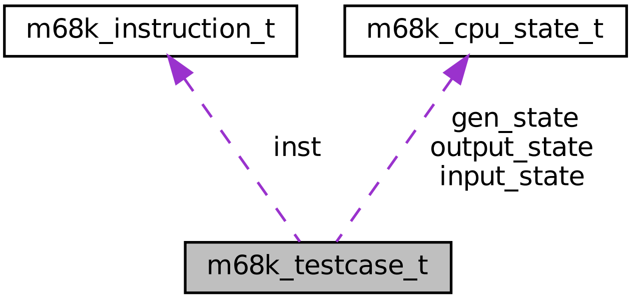
- int **mnemo**
- char **name** [8]
- int **n_words**
- uint16 **words** [1]

The documentation for this struct was generated from the following file:

- m68k-tester.cpp

10.118 m68k_testcase_t Struct Reference

Collaboration diagram for m68k_testcase_t:



Data Fields

- m68k_instruction_t ∗ **inst**
- m68k_cpu_state_t **input_state**
- m68k_cpu_state_t **output_state**
- m68k_cpu_state_t **gen_state**

The documentation for this struct was generated from the following file:

- m68k-tester.cpp

10.119 MatrixBufferType Struct Reference

Data Fields

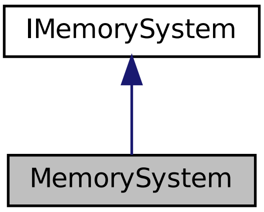
- XMMATRIX **world**
- XMMATRIX **view**
- XMMATRIX **projection**

The documentation for this struct was generated from the following file:

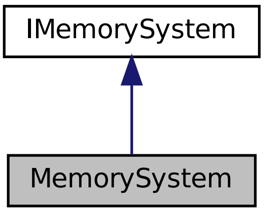
- GfxDrvDXGI.cpp

10.120 MemorySystem Class Reference

Inheritance diagram for MemorySystem:



Collaboration diagram for MemorySystem:



Public Member Functions

- UBY **ReadByte** (ULO address) override
- UWO **ReadWord** (ULO address) override
- ULO **ReadLong** (ULO address) override
- void **WriteByte** (UBY data, ULO address) override
- void **WriteWord** (UWO data, ULO address) override
- void **WriteLong** (ULO data, ULO address) override
- void **DmemSetByte** (UBY data) override
- void **DmemSetWord** (UWO data) override
- void **DmemSetLong** (ULO data) override
- void **DmemSetLongNoCounter** (ULO data, ULO offset) override
- void **DmemSetString** (const STR ∗data) override
- void **DmemSetCounter** (ULO val) override
- ULO **DmemGetCounter** () override
- void **DmemClear** () override
- void **EmemClear** () override
- void **EmemSet** (ULO index, ULO data) override

- void **EmemCardAdd** (fellow::api::vm::EmemCardInitFunc cardinit, fellow::api::vm::EmemCardMapFunc cardmap) override
- void **EmemMirror** (ULO emem_offset, UBY ∗src, ULO size) override
- void **BankSet** (fellow::api::vm::ReadByteFunc rb, fellow::api::vm::ReadWordFunc rw, fellow::api::vm::Read←LongFunc rl, fellow::api::vm::WriteByteFunc wb, fellow::api::vm::WriteWordFunc ww, fellow::api::vm::Write←LongFunc wl, UBY ∗basep, ULO bank, ULO basebank, BOOLE pointer_can_write) override
- UBY ∗ **AddressToPtr** (ULO address) override
- ULO **GetKickImageVersion** () override

The documentation for this class was generated from the following files:

- MemorySystem.h
- MemorySystem.cpp

10.121 ModuleInfo Struct Reference

Data Fields

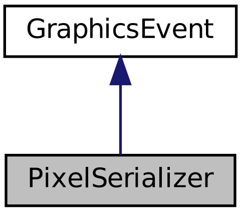
- char **filename** [MODRIP_TEMPSTRLEN]
- char **modname** [MODRIP_TEMPSTRLEN]
- char **typedesc** [MODRIP_TEMPSTRLEN]
- char **typesig** [MODRIP_TEMPSTRLEN]
- ULO **start**
- ULO **end**
- unsigned **samplesize**
- unsigned **patternsize**
- unsigned **songlength**
- unsigned **maxpattern**
- unsigned **channels**
- unsigned **instruments**

The documentation for this struct was generated from the following file:

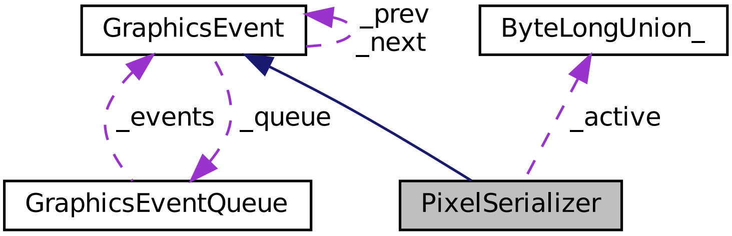
- modrip.h

10.122 PixelSerializer Class Reference

Inheritance diagram for PixelSerializer:



Collaboration diagram for PixelSerializer:



Public Member Functions

- void **Commit** (UWO dat1, UWO dat2, UWO dat3, UWO dat4, UWO dat5, UWO dat6)
- void **OutputCylindersUntil** (ULO rasterY, ULO cylinder)
- virtual void **Handler** (ULO rasterY, ULO cylinder)
- virtual void **InitializeEvent** (GraphicsEventQueue ∗queue)
- void **EndOfFrame** (void)
- void **SoftReset** (void)
- void **HardReset** (void)
- void **EmulationStart** (void)
- void **EmulationStop** (void)
- void **Startup** (void)
- void **Shutdown** (void)

Private Member Functions

- void **LogEndOfLine** (ULO rasterY, ULO cylinder)
- void **LogOutput** (ULO rasterY, ULO cylinder, ULO startCylinder, ULO untilCylinder)
- void **EventSetup** (ULO arriveTime)
- void **ShiftActive** (ULO pixelCount)
- ULO **GetOutputLine** (ULO rasterY, ULO cylinder)
- ULO **GetOutputCylinder** (ULO cylinder)
- void **SerializePixels** (ULO pixelCount)
- void **SerializeBatch** (ULO cylinderCount)

Private Attributes

- ByteLongUnion **_active** [6]
- ULO **_tmpline** [960]
- ULO **_lastCylinderOutput**
- bool **_newLine**
- bool **_activated**

Static Private Attributes

- static const ULO **FIRST_CYLINDER** = 56
- static const ULO **LAST_CYLINDER** = 25

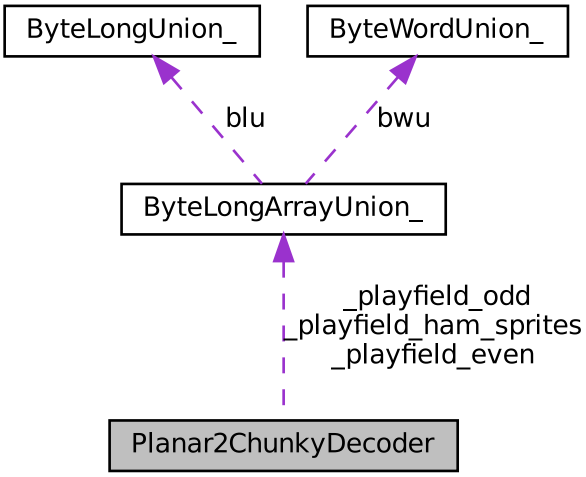
Additional Inherited Members

The documentation for this class was generated from the following files:

- PixelSerializer.h
- PixelSerializer.c

10.123 Planar2ChunkyDecoder Class Reference

Collaboration diagram for Planar2ChunkyDecoder:



Public Member Functions

- UBY ∗ **GetOddPlayfield** (void)
- UBY ∗ **GetEvenPlayfield** (void)
- UBY ∗ **GetHamSpritesPlayfield** (void)
- ULO **GetBatchSize** (void)
- void **NewBatch** (void)
- void **P2CNextPixels** (ULO pixelCount, ULO dat1, ULO dat2, ULO dat3, ULO dat4, ULO dat5, ULO dat6)
- void **P2CNext4Pixels** (ULO dat1, ULO dat2, ULO dat3, ULO dat4, ULO dat5, ULO dat6)
- void **P2CNext8Pixels** (ULO dat1, ULO dat2, ULO dat3, ULO dat4, ULO dat5, ULO dat6)

Private Member Functions

- ULO ∗ **GetEvenPlayfieldULOPtr** (void)
- ULO ∗ **GetOddPlayfieldULOPtr** (void)
- ULO **P2COdd1** (ULO dat1, ULO dat3, ULO dat5)
- ULO **P2COdd2** (ULO dat1, ULO dat3, ULO dat5)
- ULO **P2CEven1** (ULO dat2, ULO dat4, ULO dat6)
- ULO **P2CEven2** (ULO dat2, ULO dat4, ULO dat6)
- ULO **P2CDual1** (ULO dat1, ULO dat2, ULO dat3)
- ULO **P2CDual2** (ULO dat1, ULO dat2, ULO dat3)
- void **P2CNextPixelsNormal** (ULO pixelCount, ULO dat1, ULO dat2, ULO dat3, ULO dat4, ULO dat5, ULO dat6)
- void **P2CNextPixelsDual** (ULO pixelCount, ULO dat1, ULO dat2, ULO dat3, ULO dat4, ULO dat5, ULO dat6)
- void **P2CNext4PixelsNormal** (ULO dat1, ULO dat2, ULO dat3, ULO dat4, ULO dat5, ULO dat6)
- void **P2CNext4PixelsDual** (ULO dat1, ULO dat2, ULO dat3, ULO dat4, ULO dat5, ULO dat6)
- void **P2CNext8PixelsNormal** (ULO dat1, ULO dat2, ULO dat3, ULO dat4, ULO dat5, ULO dat6)
- void **P2CNext8PixelsDual** (ULO dat1, ULO dat2, ULO dat3, ULO dat4, ULO dat5, ULO dat6)

Private Attributes

- ULO **_batch_size**
- ByteLongArrayUnion **_playfield_odd**
- ByteLongArrayUnion **_playfield_even**
- ByteLongArrayUnion **_playfield_ham_sprites**

The documentation for this class was generated from the following files:

- Planar2ChunkyDecoder.h
- Planar2ChunkyDecoder.c

10.124 ptunion Union Reference

Data Fields

- ULO ∗ **lptr**
- UWO ∗ **wptr**
- UBY ∗ **bptr**
- ULO **lval**
- UWO **wval** [2]
- UBY **bval** [4]

The documentation for this union was generated from the following file:

- DEFS.H

10.125 RawDataReader Class Reference

Public Member Functions

- ULO **GetIndex** ()
- ULO **GetNextByteswappedLong** ()
- std::string **GetNextString** (ULO lengthInLongwords)
- UBY ∗ **GetNextBytes** (ULO lengthInLongwords)
- **RawDataReader** (UBY ∗rawData, ULO rawDataLength)

Private Member Functions

- void **AssertValidIndexAndLength** (ULO length)
- char **GetByteAsChar** (ULO index)
- ULO **GetByteAsLong** (ULO index)
- char **GetNextChar** ()

Private Attributes

- UBY ∗ **_rawData**
- ULO **_rawDataLength**
- ULO **_index**

The documentation for this class was generated from the following files:

- RawDataReader.h
- RawDataReader.cpp

10.126 RDB Class Reference

Public Member Functions

- void **ReadFromFile** (RDBFileReader &reader, bool geometryOnly=false)
- void **Log** ()

Data Fields

- std::string **ID**
- ULO **SizeInLongs**
- LON **CheckSum**
- ULO **HostID**
- ULO **BlockSize**
- ULO **Flags**
- ULO **BadBlockList**
- ULO **PartitionList**
- ULO **FilesystemHeaderList**
- ULO **DriveInitCode**
- ULO **Cylinders**
- ULO **SectorsPerTrack**
- ULO **Heads**
- ULO **Interleave**
- ULO **ParkingZone**
- ULO **WritePreComp**
- ULO **ReducedWrite**
- ULO **StepRate**
- ULO **RDBBlockLow**
- ULO **RDBBlockHigh**
- ULO **LowCylinder**
- ULO **HighCylinder**
- ULO **CylinderBlocks**
- ULO **AutoParkSeconds**
- ULO **HighRDSKBlock**
- std::string **DiskVendor**
- std::string **DiskProduct**
- std::string **DiskRevision**
- std::string **ControllerVendor**
- std::string **ControllerProduct**
- std::string **ControllerRevision**
- std::vector< std::unique_ptr< RDBPartition > > **Partitions**
- std::vector< std::unique_ptr< RDBFileSystemHeader > > **FileSystemHeaders**

The documentation for this class was generated from the following files:

- RDB.h
- RDB.cpp

10.127 RDBFileReader Class Reference

Public Member Functions

- std::string **ReadString** (off_t offset, size_t maxCount)
- UBY **ReadUBY** (off_t offset)
- ULO **ReadULO** (off_t offset)
- LON **ReadLON** (off_t offset)
- UBY ∗ **ReadData** (off_t offset, size_t byteCount)
- **RDBFileReader** (FILE ∗F)

Private Attributes

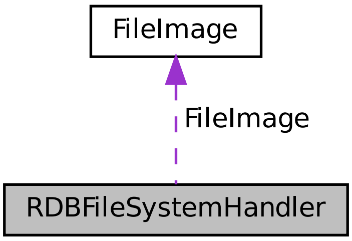
- FILE ∗ **_F**

The documentation for this class was generated from the following files:

- RDBFileReader.h
- RDBFileReader.cpp

10.128 RDBFileSystemHandler Struct Reference

Collaboration diagram for RDBFileSystemHandler:



Public Member Functions

- bool **ReadFromFile** (RDBFileReader &reader, ULO blockChainStart, ULO blockSize)

Data Fields

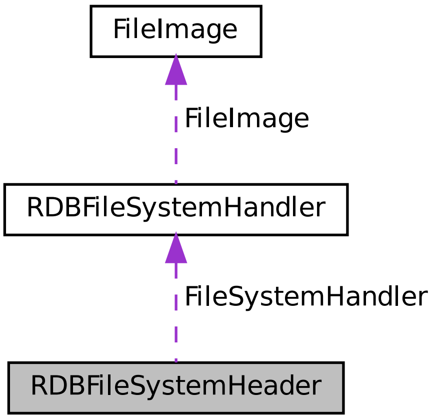
- ULO **Size**
- std::unique_ptr< UBY > **RawData**
- fellow::hardfile::hunks::FileImage **FileImage**

The documentation for this struct was generated from the following files:

- RDBFileSystemHandler.h
- RDBFileSystemHandler.cpp

10.129 RDBFileSystemHeader Class Reference

Collaboration diagram for RDBFileSystemHeader:



Public Member Functions

- void **ReadFromFile** (RDBFileReader &reader, ULO blockChainStart, ULO blockSize)
- void **Log** ()

Data Fields

- ULO **SizeInLongs**
- LON **CheckSum**
- ULO **HostID**
- ULO **Next**
- ULO **Flags**
- ULO **DOSType**
- ULO **Version**
- ULO **PatchFlags**
- ULO **DnType**
- ULO **DnTask**
- ULO **DnLock**
- ULO **DnHandler**
- ULO **DnStackSize**
- ULO **DnPriority**
- ULO **DnStartup**
- ULO **DnSegListBlock**
- ULO **DnGlobalVec**
- ULO **Reserved2** [23]
- RDBFileSystemHandler **FileSystemHandler**

The documentation for this class was generated from the following files:

- RDBFileSystemHeader.h
- RDBFileSystemHeader.cpp

10.130 RDBHandler Class Reference

Static Public Member Functions

- static bool **HasRigidDiskBlock** (RDBFileReader &reader)
- static RDB ∗ **GetDriveInformation** (RDBFileReader &reader, bool geometryOnly=false)

The documentation for this class was generated from the following files:

- RDBHandler.h
- RDBHandler.cpp

10.131 RDBLSegBlock Struct Reference

Public Member Functions

- LON **GetDataSize** () const
- const UBY ∗ **GetData** () const
- void **ReadFromFile** (RDBFileReader &reader, ULO index)
- void **Log** ()

Data Fields

- std::string **ID**
- LON **Blocknumber**
- LON **SizeInLongs**
- LON **CheckSum**
- LON **HostID**
- LON **Next**
- std::unique_ptr< const UBY > **Data**

The documentation for this struct was generated from the following files:

- RDBLSegBlock.h
- RDBLSegBlock.cpp

10.132 RDBPartition Struct Reference

Public Member Functions

- bool **IsAutomountable** ()
- bool **IsBootable** ()
- void **ReadFromFile** (RDBFileReader &reader, ULO blockChainStart, ULO blockSize)
- void **Log** ()

Data Fields

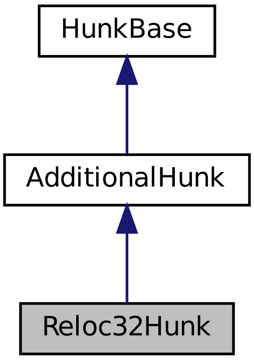
- std::string **ID**
- ULO **SizeInLongs**
- ULO **CheckSum**
- ULO **HostID**
- ULO **Next**
- ULO **Flags**
- ULO **BadBlockList**
- ULO **DevFlags**
- char **DriveNameLength**
- std::string **DriveName**
- ULO **SizeOfVector**
- ULO **SizeBlock**
- ULO **SecOrg**
- ULO **Surfaces**
- ULO **SectorsPerBlock**
- ULO **BlocksPerTrack**
- ULO **Reserved**
- ULO **PreAlloc**
- ULO **Interleave**
- ULO **LowCylinder**
- ULO **HighCylinder**
- ULO **NumBuffer**
- ULO **BufMemType**
- ULO **MaxTransfer**
- ULO **Mask**
- ULO **BootPri**
- ULO **DOSType**
- ULO **Baud**
- ULO **Control**
- ULO **Bootblocks**

The documentation for this struct was generated from the following files:

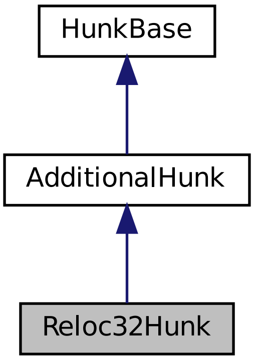
- RDBPartition.h
- RDBPartition.cpp

10.133 Reloc32Hunk Class Reference

Inheritance diagram for Reloc32Hunk:



Collaboration diagram for Reloc32Hunk:



Public Member Functions

- ULO **GetID** () override
- ULO **GetOffsetTableCount** ()
- Reloc32OffsetTable ∗ **GetOffsetTable** (ULO index)
- void **Parse** (RawDataReader &rawDataReader) override
- **Reloc32Hunk** (ULO sourceHunkIndex)

Private Attributes

- std::vector< std::unique_ptr< Reloc32OffsetTable > > **_offsetTables**

Static Private Attributes

- static const ULO **ID** = Reloc32HunkID

The documentation for this class was generated from the following files:

- Reloc32Hunk.h
- Reloc32Hunk.cpp

10.134 Reloc32OffsetTable Class Reference

Public Member Functions

- ULO **GetRelatedHunkIndex** ()
- ULO **GetOffsetCount** ()
- ULO **GetOffset** (ULO index)
- void **Parse** (RawDataReader &rawDataReader, ULO offsetCount)
- **Reloc32OffsetTable** (ULO relatedHunkIndex)

Private Attributes

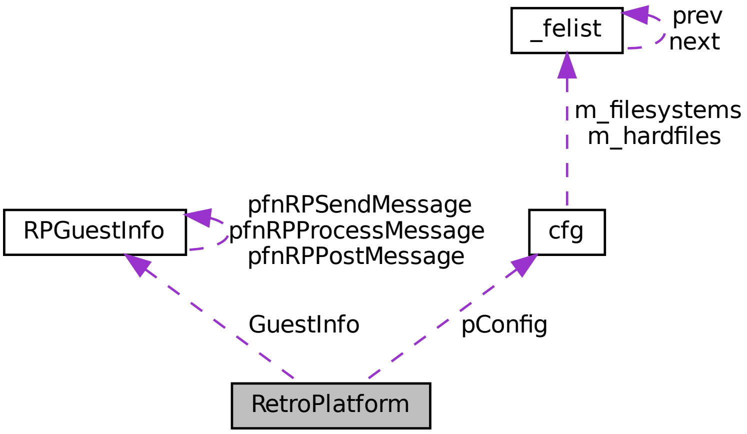
- ULO **_relatedHunkIndex**
- std::vector< ULO > **_offsets**

The documentation for this class was generated from the following files:

- Reloc32OffsetTable.h
- Reloc32OffsetTable.cpp

10.135 RetroPlatform Class Reference

Collaboration diagram for RetroPlatform:



Public Member Functions

- void **EmulationStart** (void)
- void **EmulationStop** (void)
- void EnterHeadlessMode (void)
- ULO **GetClippingOffsetLeftAdjusted** (void)
- ULO **GetClippingOffsetTopAdjusted** (void)
- ULO **GetDisplayScale** (void)
- bool **GetEmulationPaused** (void)
- ULO **GetEscapeKey** (void)
- ULONGLONG **GetEscapeKeyHeldSince** (void)
- ULO **GetEscapeKeyHoldTime** (void)
- ULONGLONG **GetEscapeKeySimulatedTargetTime** (void)
- bool GetHeadlessMode (void)
- HWND **GetParentWindowHandle** (void)
- bool **GetScanlines** (void)
- ULO **GetScreenHeightAdjusted** (void)
- ULO **GetScreenWidthAdjusted** (void)
- ULO **GetSourceBufferWidth** (void)
- ULO **GetSourceBufferHeight** (void)
- ULONGLONG GetTime (void)
- bool PostEscaped (void)
- bool PostFloppyDriveLED (const ULO, const bool, const bool)
- bool PostFloppyDriveSeek (const ULO, const ULO)
- bool PostGameportActivity (const ULO, const ULO)
- bool PostHardDriveLED (const ULO, const bool, const bool)
- bool **SendActivated** (const bool, const LPARAM)
- bool SendClose (void)
- bool SendEnable (const bool)
- bool SendFloppyDriveContent (const ULO, const STR ∗, const bool)
- bool SendHardDriveContent (const ULO, const STR ∗, const bool)
- bool SendFloppyDriveReadOnly (const ULO, const bool)
- bool SendFloppyTurbo (const bool)
- bool **SendMouseCapture** (const bool)
- void SetClippingOffsetLeft (const ULO)
- void SetClippingOffsetTop (const ULO)
- void SetEscapeKey (const ULO)
- ULONGLONG **SetEscapeKeyHeld** (const bool)
- void **SetEscapeKeyHoldTime** (const ULO)
- void **SetEscapeKeySimulatedTargetTime** (const ULONGLONG)
- void **SetHeadlessMode** (const bool)
- void **SetHostID** (const char ∗)
- void SetScreenHeight (ULO)
- void **SetScreenMode** (const char ∗)
- void SetScreenWidth (ULO)
- void **SetWindowInstance** (HINSTANCE)
- void **RegisterRetroPlatformScreenMode** (const bool bStartup)
- LRESULT CALLBACK HostMessageFunction (UINT, WPARAM, LPARAM, LPCVOID, DWORD, LPARAM)
- BOOL FAR PASCAL EnumerateJoystick (LPCDIDEVICEINSTANCE pdinst, LPVOID pvRef)
- void **Shutdown** (void)
- void **Startup** (void)

Private Member Functions

- bool CheckEmulationNecessities (void)
- bool ConnectInputDeviceToPort (const ULO, const ULO, DWORD, const STR ∗)
- void DetermineScreenModeFromConfig (struct RPScreenMode ∗, cfg ∗)
- int EnumerateJoysticks (void)
- ULO **GetClippingOffsetLeft** (void)
- ULO **GetClippingOffsetTop** (void)
- ULO **GetCPUSpeed** (void)
- bool GetHostVersion (ULO ∗, ULO ∗, ULO ∗)
- const STR ∗ GetMessageText (ULO)
- ULO **GetScreenHeight** (void)
- ULO **GetScreenWidth** (void)
- bool **GetScreenWindowed** (void)
- bool PostMessageToHost (ULO, WPARAM, LPARAM, const RPGUESTINFO ∗)
- bool PostPowerLEDIntensityPercent (const WPARAM)
- bool SendMessageToHost (ULO, WPARAM, LPARAM, LPCVOID, DWORD, const RPGUESTINFO ∗, LR↩ ESULT ∗)
- void SetCustomKeyboardLayout (const ULO, const STR ∗)
- void **SetDisplayScale** (const ULO)
- void **SetEmulationPaused** (const bool)
- void **SetEmulationState** (const bool)
- void **SetEmulatorQuit** (const bool)
- void **SetScanlines** (const bool)
- void **SetScreenModeStruct** (struct RPScreenMode ∗)
- void **SetScreenWindowed** (const bool)
- bool SendEnabledFloppyDrives (void)
- bool SendEnabledHardDrives (void)
- bool SendFeatures (void)
- bool **SendGameports** (const ULO)
- bool SendInputDevice (const DWORD, const DWORD, const DWORD, const WCHAR ∗, const WCHAR ∗)
- bool SendInputDevices (void)
- bool SendScreenMode (HWND)

Private Attributes

- STR **szHostID** [CFG_FILENAME_LENGTH]
- bool bRetroPlatformMode = false

  host ID that was passed over by the RetroPlatform player
- bool bInitialized = false

  flag to indicate that emulator operates in RetroPlatform/"headless" mode
- bool **bEmulationState** = false
- bool **bEmulationPaused** = false
- bool **bEmulatorQuit** = false
- bool **bMouseCaptureRequestedByHost** = false
- ULO **lMainVersion** = 0
- ULO **lRevision** = 0
- ULO **lBuild** = 0
- LON **lClippingOffsetLeftRP** = RETRO_PLATFORM_OFFSET_ADJUST_LEFT
- LON **lClippingOffsetTopRP** = RETRO_PLATFORM_OFFSET_ADJUST_TOP
- LON **lScreenWidthRP** = 0
- LON **lScreenHeightRP** = 0
- ULO **lScreenMode** = 0

- bool **bScreenWindowed** = true
- ULO **lDisplayScale** = 1
- bool **bScanlines** = false
- ULO **lEscapeKey** = 1
- ULO **lEscapeKeyHoldTime** = 600
- ULONGLONG **lEscapeKeyHeldSince** = 0
- ULONGLONG **lEscapeKeySimulatedTargetTime** = 0
- int **iNumberOfJoysticksAttached** = 0
- RPGUESTINFO **GuestInfo**
- HINSTANCE **hWindowInstance** = NULL
- HWND **hGuestWindow** = NULL
- cfg ∗ pConfig

  RetroPlatform copy of WinFellow configuration.

10.135.1 Member Function Documentation

10.135.1.1 CheckEmulationNecessities()

```
bool CheckEmulationNecessities (
            void )  [private]
```

Verifies that the prerequisites to start the emulation are available.

Validates that the configuration contains a path to a Kickstart ROM, and that the file can be opened successfully for reading.

Returns

true, when Kickstart ROM can be opened successfully for reading; false otherwise

10.135.1.2 ConnectInputDeviceToPort()

```
bool ConnectInputDeviceToPort (
            const ULO lGameport,
            const ULO lDeviceType,
            DWORD dwFlags,
            const STR * szName )  [private]
```

Attach input devices to gameports during runtime of the emulator.

The device is selected in the RetroPlatform player and passed to the emulator in form of an IPC message.

10.135.1.3 DetermineScreenModeFromConfig()

```
void DetermineScreenModeFromConfig (
            struct RPScreenMode * RetroPlatformScreenMode,
            cfg * RetroPlatformConfig )  [private]
```

Translate the screenmode configured in the configuration file and pass it along to the RetroPlatform Player.

10.135.1.4 EnterHeadlessMode()

```
void EnterHeadlessMode (
             void )
```

The main control function when operating in RetroPlatform headless mode.

This function performs the start of the emulator session. On a reset event, winDrvEmulationStart will exit without bRetroPlatformEmulatorQuit being set.

10.135.1.5 EnumerateJoystick()

```
BOOL FAR PASCAL EnumerateJoystick (
             LPCDIDEVICEINSTANCE pdinst,
             LPVOID pvRef )
```

Joystick enumeration function.

10.135.1.6 EnumerateJoysticks()

```
int EnumerateJoysticks (
             void ) [private]
```

Determine the number of joysticks connected to the system.

10.135.1.7 GetHeadlessMode()

```
bool GetHeadlessMode (
             void )
```

Verify if the emulator is operating in RetroPlatform mode.

Checks the value of the bRetroPlatformMode flag. It is set to true, if a RetroPlatform host ID has been passed along as a commandline parameter.

Returns

true if WinFellow was called from Cloanto RetroPlatform, false if not.

10.135.1.8 GetHostVersion()

```
bool GetHostVersion (
             ULO * lpMainVersion,
             ULO * lpRevision,
             ULO * lpBuild ) [private]
```

Determine the RetroPlatform host version.

Parameters

out	*lpMainVersion*	main version number
out	*lpRevision*	revision number
out	*lpBuild*	build number

Returns

true is successful, false otherwise.

10.135.1.9 GetMessageText()

```
const STR * GetMessageText (
              ULO iMsg )  [private]
```

Translate a RetroPlatform IPC message code into readable text.

10.135.1.10 GetTime()

```
ULONGLONG GetTime (
              void  )
```

Determine a timestamp for the current time.

10.135.1.11 HostMessageFunction()

```
LRESULT CALLBACK HostMessageFunction (
              UINT uMessage,
              WPARAM wParam,
              LPARAM lParam,
              LPCVOID pData,
              DWORD dwDataSize,
              LPARAM lMsgFunctionParam )
```

host message function that is used as callback to receive IPC messages from the host.

10.135.1.12 PostEscaped()

```
bool PostEscaped (
              void  )
```

Post message to the player to signalize that the guest wants to escape the mouse cursor.

10.135.1.13 PostFloppyDriveLED()

```
bool PostFloppyDriveLED (
              const ULO lFloppyDriveNo,
              const bool bMotorActive,
              const bool bWriteActivity )
```

Control status of the RetroPlatform floppy drive LEDs.

Sends LED status changes to the RetroPlatform host in the form of RP_IPC_TO_HOST_DEVICEACTIVITY messages, so that floppy read and write activity can be displayed, and detected (undo functionality uses write messages as fallback method to detect changed floppy images).

Parameters

in	*lFloppyDriveNo*	floppy drive index (0-3)
in	*bMotorActive*	state of floppy drive motor (active/inactive)
in	*bWriteActivity*	type of access (write/read)

Returns

true if message sent successfully, false otherwise.

10.135.1.14 PostFloppyDriveSeek()

```
bool PostFloppyDriveSeek (
            const ULO lFloppyDriveNo,
            const ULO lTrackNo )
```

Send floppy drive seek events to RetroPlatform host.

Will notify the RetroPlatform player about changes in the drive head position.

Parameters

in	*lFloppyDriveNo*	index of floppy drive
in	*lTrackNo*	index of floppy track

Returns

true if message sent successfully, false otherwise.

10.135.1.15 PostGameportActivity()

```
bool PostGameportActivity (
            const ULO lGameport,
            const ULO lGameportMask )
```

Send gameport activity to RetroPlatform host.

10.135.1.16 PostHardDriveLED()

```
bool PostHardDriveLED (
            const ULO lHardDriveNo,
            const bool bActive,
            const bool bWriteActivity )
```

Control status of the RetroPlatform hard drive LEDs.

Sends LED status changes to the RetroPlatform host in the form of RP_IPC_TO_HOST_DEVICEACTIVITY messages, so that hard drive read and write activity can be displayed, and detected (undo functionality uses write messages as fallback method to detect changed floppy images).

Parameters

in	*lHardDriveNo*	hard drive index (0-...)
in	*bActive*	flag indicating disk access (active/inactive)
in	*bWriteActivity*	flag indicating type of access (write/read)

Returns

true if message sent successfully, false otherwise.

10.135.1.17 PostMessageToHost()

```
bool PostMessageToHost (
            ULO iMessage,
            WPARAM wParam,
            LPARAM lParam,
            const RPGUESTINFO * pGuestInfo )  [private]
```

Asynchronously post a message to the RetroPlatform host.

A message is posted to the host asynchronously, i.e. without waiting for results.

10.135.1.18 PostPowerLEDIntensityPercent()

```
bool PostPowerLEDIntensityPercent (
            const WPARAM wIntensityPercent )  [private]
```

Control status of power LED in RetroPlatform player.

Examines the current on/off state of the emulator session and sends it to the RetroPlatform player.

Parameters

in	*wIntensityPercent*	intensity of the power LED in percent, with 0 being off, 100 being full intensity.

Returns

true, if valid value was passed, false if invalid value.

10.135.1.19 SendClose()

```
bool SendClose (
            void  )
```

Notify the player that the user request to close the emulation session.

The player will examine if changes to the package were performed that require user feedback (media changed where undo is enabled, parameters like e.g. clipping were changed, ...). The user can choose what to commit and proceed with quitting, or cancel. The player can then either notify the emulator to quit via an IPC message RP_IPC_TO_GUEST_CLOSE, or do nothing and let the session continue.

10.135.1.20 SendEnable()

```
bool SendEnable (
            const bool bEnabled )
```

Send enable/disable messages to the RetroPlatform player.

These are sent on WM_ENABLE messages.

10.135.1.21 SendEnabledFloppyDrives()

```
bool SendEnabledFloppyDrives (
            void ) [private]
```

Send list of enabled floppy drives to the RetroPlatform host.

An RP_IPC_TO_HOST_DEVICES message is sent to the host, indicating the floppy drives enabled in the guest. Must be called after the activation of the config, and before sending the screen mode.

Returns

true if message was sent successfully, false otherwise.

10.135.1.22 SendEnabledHardDrives()

```
bool SendEnabledHardDrives (
            void ) [private]
```

Send list of enabled hard drives to the RetroPlatform host.

An RP_IPC_TO_HOST_DEVICES message is sent to the host, indicating the hard drives enabled in the guest. Must be called after the activation of the config, and before sending the screen mode.

Returns

true if message was sent successfully, false otherwise.

10.135.1.23 SendFeatures()

```
bool SendFeatures (
            void ) [private]
```

Send list of features supported by the guest to the RetroPlatform host.

An RP_IPC_TO_HOST_FEATURES message is sent to the host, with flags indicating the features supported by the guest.

Returns

true if message was sent successfully, false otherwise.

10.135.1.24 SendFloppyDriveContent()

```
bool SendFloppyDriveContent (
            const ULO lFloppyDriveNo,
            const STR * szImageName,
            const bool bWriteProtected )
```

Send content of floppy drive to RetroPlatform host. The read-only state is determined and sent here, however at this point it is usually wrong, as floppySetDiskImage only reflects the ability to write to the file in the writeprot flag. The actual state within the config is configured in a separate call within cfgManagerConfigurationActivate - therefore an update message is sent later.

Parameters

in	*lFloppyDriveNo*	floppy drive index (0-3)
in	*szImageName*	ANSI string containing the floppy image name
in	*bWriteProtected*	flag indicating the read-only state of the drive

Returns

true if message sent successfully, false otherwise.

See also

RetroPlatformSendFloppyDriveReadOnly

10.135.1.25 SendFloppyDriveReadOnly()

```
bool SendFloppyDriveReadOnly (
            const ULO lFloppyDriveNo,
            const bool bWriteProtected )
```

Send actual write protection state of drive to RetroPlatform host. Ignores drives that are not enabled.

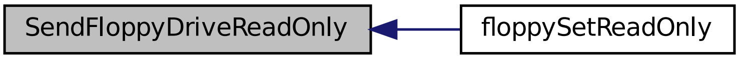
Parameters

in	*lFloppyDriveNo*	floppy drive index (0-3)
in	*bWriteProtected*	flag indicating the read-only state of the drive

Returns

true if message sent successfully, false otherwise.

Here is the caller graph for this function:



10.135.1.26 SendFloppyTurbo()

```
bool SendFloppyTurbo (
             const bool bTurbo )
```

Send floppy turbo mode state to RetroPlatform host.

Parameters

in	*bTurbo*	flag indicating state of turbo mode

Returns

true if message sent successfully, false otherwise.

10.135.1.27 SendHardDriveContent()

```
bool SendHardDriveContent (
             const ULO lHardDriveNo,
             const STR * szImageName,
             const bool bWriteProtected )
```

Send content of hard drive to RetroPlatform host.

Parameters

in	*lHardDriveNo*	hard drive index (0-...)
in	*szImageName*	ANSI string containing the floppy image name
in	*bWriteProtected*	flag indicating the read-only state of the drive

Returns

true if message sent successfully, false otherwise.

10.135.1.28 SendInputDevice()

```
bool SendInputDevice (
            const DWORD dwHostInputType,
            const DWORD dwInputDeviceFeatures,
            const DWORD dwFlags,
            const WCHAR * szHostInputID,
            const WCHAR * szHostInputName )  [private]
```

Send a single input device to the RetroPlatform player.

10.135.1.29 SendInputDevices()

```
bool SendInputDevices (
            void  )  [private]
```

Send list of available input device options to the RetroPlatform player.

The emulator is supposed to enumerate the Windows devices and identify them via unique IDs; joysticks are sent after enumeration during emulator session start, other devices are sent here

10.135.1.30 SendMessageToHost()

```
bool SendMessageToHost (
            ULO iMessage,
            WPARAM wParam,
            LPARAM lParam,
            LPCVOID pData,
            DWORD dwDataSize,
            const RPGUESTINFO * pGuestInfo,
            LRESULT * plResult )  [private]
```

Send an IPC message to RetroPlatform host.

Returns

true is sucessfully sent, false otherwise.

10.135.1.31 SendScreenMode()

```
bool SendScreenMode (
             HWND hWnd )  [private]
```

Send screen mode to the player.

This step finalizes the transfer of guest features to the player and will enable the emulation.

10.135.1.32 SetClippingOffsetLeft()

```
void SetClippingOffsetLeft (
             const ULO lOffsetLeft )
```

Set clipping offset that is applied to the left of the picture.

10.135.1.33 SetClippingOffsetTop()

```
void SetClippingOffsetTop (
             const ULO lOffsetTop )
```

Set clipping offset that is applied to the top of the picture

10.135.1.34 SetCustomKeyboardLayout()

```
void SetCustomKeyboardLayout (
             const ULO lGameport,
             const STR * pszKeys )  [private]
```

configure keyboard layout to custom key mappings

Gameport 0 is statically mapped to internal keyboard layout GP_JOYKEY0, gameport 1 to GP_JOYKEY1 as we reconfigure them anyway

10.135.1.35 SetEscapeKey()

```
void SetEscapeKey (
             const ULO lNewEscapeKey )
```

Set RetroPlatform escape key.

Called during parsing of the command-line parameters, which is why the keyboard modules have to be initialized before the config modules, as we use the key mappings here.

10.135.1.36 SetScreenHeight()

```
void SetScreenHeight (
             ULO lHeight )
```

Set screen height.

10.135.1.37 SetScreenWidth()

```
void SetScreenWidth (
            ULO lWidth )
```

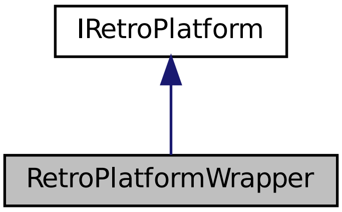
Set screen width.

The documentation for this class was generated from the following files:

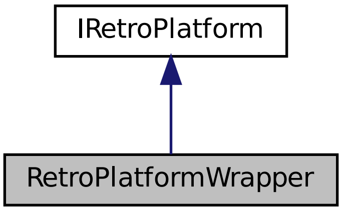
- RetroPlatform.h
- RetroPlatform.cpp

10.136 RetroPlatformWrapper Class Reference

Inheritance diagram for RetroPlatformWrapper:



Collaboration diagram for RetroPlatformWrapper:



Public Member Functions

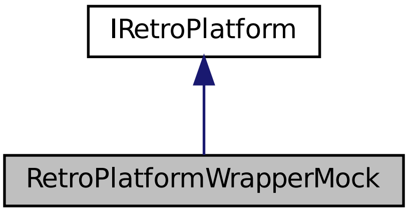
- bool **SendHardDriveContent** (const ULO lHardDriveNo, const STR ∗szImageName, const bool bWrite←Protected) override
- bool **PostHardDriveLED** (const ULO lHardDriveNo, const bool bActive, const bool bWriteActivity) override

The documentation for this class was generated from the following files:

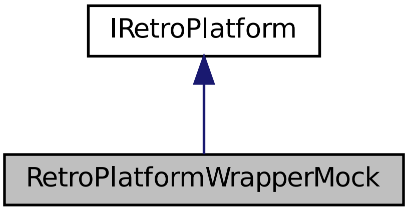
- RetroPlatformWrapper.h
- RetroPlatformWrapper.cpp

10.137 RetroPlatformWrapperMock Class Reference

Inheritance diagram for RetroPlatformWrapperMock:



Collaboration diagram for RetroPlatformWrapperMock:



Public Member Functions

- bool **SendHardDriveContent** (const ULO lHardDriveNo, const STR ∗szImageName, const bool bWrite↩ Protected) override
- bool **PostHardDriveLED** (const ULO lHardDriveNo, const bool bActive, const bool bWriteActivity) override

The documentation for this class was generated from the following files:

- RetroPlatformWrapperMock.h
- RetroPlatformWrapperMock.cpp

10.138 RPDeviceContent Struct Reference

Data Fields

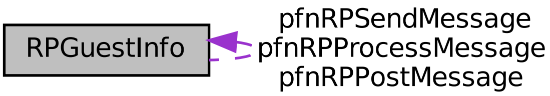
- BYTE **btDeviceCategory**
- BYTE **btDeviceNumber**
- DWORD **dwInputDevice**
- DWORD **dwFlags**
- WCHAR **szContent** [260]

The documentation for this struct was generated from the following file:

- RetroPlatformIPC.h

10.139 RPGuestInfo Struct Reference

Collaboration diagram for RPGuestInfo:



Data Fields

- HINSTANCE **hInstance**
- HWND **hHostMessageWindow**
- HWND **hGuestMessageWindow**
- BOOL **bGuestClassRegistered**
- PFN_MsgFunction **pfnMsgFunction**
- LPARAM **lMsgFunctionParam**
- HMODULE **hRPGuestDLL**
- LPVOID **pRPGuestDLLData**
- PFN_RPProcessMessage **pfnRPProcessMessage**
- PFN_RPSendMessage **pfnRPSendMessage**
- PFN_RPPostMessage **pfnRPPostMessage**

The documentation for this struct was generated from the following file:

- RetroPlatformGuestIPC.h

10.140 RPInputDeviceDescription Struct Reference

Data Fields

- DWORD **dwHostInputType**
- WCHAR **szHostInputID** [260]
- WCHAR **szHostInputName** [260]
- DWORD **dwHostInputVendorID**
- DWORD **dwHostInputProductID**
- DWORD **dwInputDeviceFeatures**
- DWORD **dwFlags**

The documentation for this struct was generated from the following file:

- RetroPlatformIPC.h

10.141 RPScreenCapture Struct Reference

Data Fields

- DWORD **dwFlags**
- WCHAR **szScreenRaw** [260]
- WCHAR **szScreenFiltered** [260]

The documentation for this struct was generated from the following file:

- RetroPlatformIPC.h

10.142 RPScreenMode Struct Reference

Data Fields

- DWORD **dwScreenMode**
- LONG **lClipLeft**
- LONG **lClipTop**
- LONG **lClipWidth**
- LONG **lClipHeight**
- HWND **hGuestWindow**
- DWORD **dwClipFlags**
- LONG **lTargetWidth**
- LONG **lTargetHeight**

The documentation for this struct was generated from the following file:

- RetroPlatformIPC.h

10.143 RtcOkiMsm6242rs Class Reference

Public Member Functions

- UWO **read** (ULO address)
- void **write** (UWO data, ULO address)
- void **logRtcTime** (STR ∗msg)

Private Member Functions

- struct tm ∗ **GetCurrentOrHeldTime** (void)
- void **SetCurrentTime** (struct tm ∗datetime)
- int **GetRegisterNumberFromAddress** (ULO address)
- UWO **GetFirstDigit** (int value)
- void **SetFirstDigit** (struct tm &datetime, int &value, UWO data)
- void **ReplaceFirstDigit** (int &value, int new_digit)
- UWO **GetSecondDigit** (int value)
- void **SetSecondDigit** (struct tm &datetime, int &value, UWO data)
- void **ReplaceSecondDigit** (int &value, int new_digit)
- void **ReplaceSecondDigitAllowBCDOverflow** (int &value, int new_digit)
- UWO **GetSecondRegister** (void)
- void **SetSecondRegister** (UWO data)
- UWO **GetTenSecondRegister** (void)
- void **SetTenSecondRegister** (UWO data)
- UWO **GetMinuteRegister** (void)
- void **SetMinuteRegister** (UWO data)
- UWO **GetTenMinuteRegister** (void)
- void **SetTenMinuteRegister** (UWO data)
- UWO **GetHourRegister** (void)
- void **SetHourRegister** (UWO data)
- UWO **GetTenHourRegister** (void)
- void **SetTenHourRegister** (UWO data)
- UWO **GetDayRegister** (void)
- void **SetDayRegister** (UWO data)
- UWO **GetTenDayRegister** (void)
- void **SetTenDayRegister** (UWO data)
- UWO **GetMonthRegister** (void)
- void **SetMonthRegister** (UWO data)
- UWO **GetTenMonthRegister** (void)
- void **SetTenMonthRegister** (UWO data)
- UWO **GetYearRegister** (void)
- void **SetYearRegister** (UWO data)
- UWO **GetTenYearRegister** (void)
- void **SetTenYearRegister** (UWO data)
- UWO **GetWeekRegister** (void)
- void **SetWeekRegister** (UWO data)
- UWO **GetControlRegisterD** (void)
- void **SetControlRegisterD** (UWO data)
- UWO **GetControlRegisterE** (void)
- void **SetControlRegisterE** (UWO data)
- UWO **GetControlRegisterF** (void)
- void **SetControlRegisterF** (UWO data)
- void **InitializeRegisterGetters** (void)
- void **InitializeRegisterSetters** (void)

Private Attributes

- RtcOkiMsm6242rsRegisterGetter **_registerGetters** [16]
- RtcOkiMsm6242rsRegisterSetter **_registerSetters** [16]
- time_t **_rtcLastActualTime**
- time_t **_rtcTime**
- int **_rtcWeekdayModifier**
- UWO **_irqFlag**
- UWO **_holdFlag**
- UWO **_thirtySecAdjFlag**
- UWO **_busyFlag**
- UWO **_maskFlag**
- UWO **_itrptStdFlag**
- UWO **_t0Flag**
- UWO **_t1Flag**
- UWO **_restFlag**
- UWO **_stopFlag**
- UWO **_twentyFourTwelveFlag**
- UWO **_testFlag**

The documentation for this class was generated from the following files:

- RtcOkiMsm6242rs.h
- RtcOkiMsm6242rs.cpp

10.144 Script Class Reference

Public Member Functions

- void **RecordKey** (UBY keyCode)
- void **RecordMouse** (gameport_inputs mousedev, LON x, LON y, BOOLE button1, BOOLE button2, BOOLE button3)
- void **RecordJoystick** (gameport_inputs joydev, BOOLE left, BOOLE up, BOOLE right, BOOLE down, BO↩ OLE button1, BOOLE button2)
- void **RecordEmulatorAction** (kbd_event action)
- void **ExecuteUntil** (ULL frameNumber, ULO lineNumber)
- void **Load** (const string &filename)
- void **Save** (const string &filename)

Private Member Functions

- string **GetStringForAction** (kbd_event action)
- UBY **GetIdForAction** (const string &action)
- void **ExecuteMouseCommand** (const string &parameters)
- void **ExecuteKeyCommand** (const string &parameters)
- void **ExecuteJoystickCommand** (const string &parameters)
- void **ExecuteEmulatorActionCommand** (const string &parameters)
- void **Execute** (const ScriptLine &line)

Private Attributes

- const char ∗ **KeyCommand** = "Key"
- const char ∗ **MouseCommand** = "Mouse"
- const char ∗ **JoystickCommand** = "Joystick"
- const char ∗ **EmulatorActionCommand** = "EmulatorAction"
- unsigned int **_nextLine**
- vector< ScriptLine > **_lines**
- bool **_record**

The documentation for this class was generated from the following files:

- Script.h
- Script.cpp

10.145 ScriptLine Struct Reference

Public Member Functions

- **ScriptLine** (ULL frameNumber, ULO lineNumber, const string &command, const string &parameters)

Data Fields

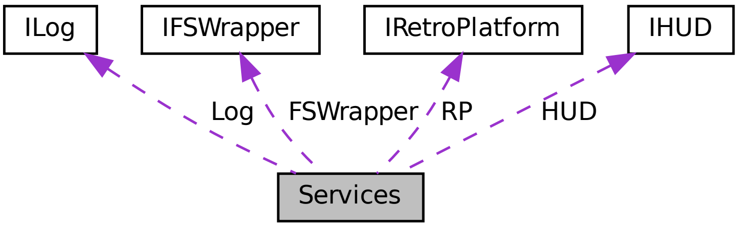
- ULL **FrameNumber**
- ULO **LineNumber**
- string **Command**
- string **Parameters**

The documentation for this struct was generated from the following files:

- Script.h
- Script.cpp

10.146 Services Class Reference

Collaboration diagram for Services:



Public Member Functions

- **Services** (fellow::api::service::IHUD &hud, fellow::api::service::IFSWrapper &fsWrapper, fellow::api::service::ILog &log, fellow::api::service::IRetroPlatform &retroPlatform)

Data Fields

- fellow::api::service::IHUD & **HUD**
- fellow::api::service::IFSWrapper & **FSWrapper**
- fellow::api::service::ILog & **Log**
- fellow::api::service::IRetroPlatform & **RP**

The documentation for this class was generated from the following file:

- Services.h

10.147 sound_device Struct Reference

Data Fields

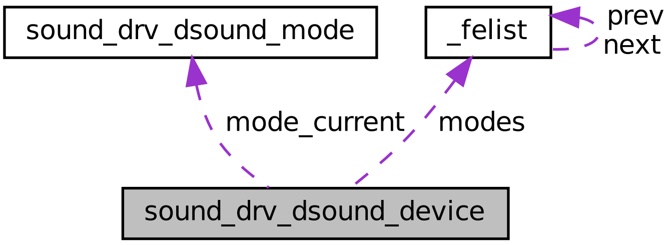
- BOOLE **mono**
- BOOLE **stereo**
- BOOLE **bits8**
- BOOLE **bits16**
- ULO **rates_max** [2][2]

The documentation for this struct was generated from the following file:

- SOUND.H

10.148 sound_drv_dsound_device Struct Reference

Collaboration diagram for sound_drv_dsound_device:



Data Fields

- LPDIRECTSOUND **lpDS**
- LPDIRECTSOUNDBUFFER **lpDSB**
- LPDIRECTSOUNDBUFFER **lpDSBS**
- LPDIRECTSOUNDNOTIFY **lpDSN**
- felist ∗ **modes**
- sound_drv_dsound_mode ∗ **mode_current**
- HANDLE **notifications** [3]
- HANDLE **data_available**
- HANDLE **can_add_data**
- HANDLE **mutex**
- UWO ∗ **pending_data_left**
- UWO ∗ **pending_data_right**
- ULO **pending_data_sample_count**
- HANDLE **thread**
- DWORD **thread_id**
- bool **notification_supported**
- ULO **mmtimer**
- ULO **mmresolution**
- DWORD **lastreadpos**

The documentation for this struct was generated from the following file:

- SOUNDDRV.C

10.149 sound_drv_dsound_mode Struct Reference

Data Fields

- ULO **rate**
- bool **bits16**
- bool **stereo**
- ULO **buffer_sample_count**
- ULO **buffer_block_align**

The documentation for this struct was generated from the following file:

- SOUNDDRV.C

10.150 spr_action_list_item Struct Reference

Data Fields

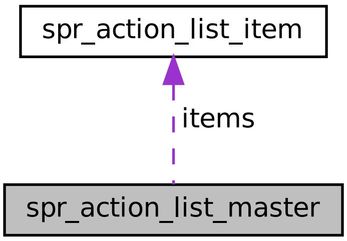
- ULO **raster_y**
- ULO **raster_x**
- spr_register_func **called_function**
- UWO **data**
- ULO **address**

The documentation for this struct was generated from the following file:

- LineExactSprites.h

10.151 spr_action_list_master Struct Reference

Collaboration diagram for spr_action_list_master:



Data Fields

- ULO **count**
- spr_action_list_item **items** [SPRITE_MAX_LIST_ITEMS]

The documentation for this struct was generated from the following file:

- LineExactSprites.h

10.152 spr_merge_list_item Struct Reference

Data Fields

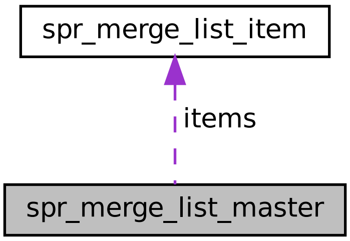
- UBY **sprite_data** [16]
- ULO **sprx**

The documentation for this struct was generated from the following file:

- LineExactSprites.h

10.153 spr_merge_list_master Struct Reference

Collaboration diagram for spr_merge_list_master:



Data Fields

- ULO **count**
- spr_merge_list_item **items** [SPRITE_MAX_LIST_ITEMS]

The documentation for this struct was generated from the following file:

- LineExactSprites.h

10.154 sprham24helper Union Reference

Data Fields

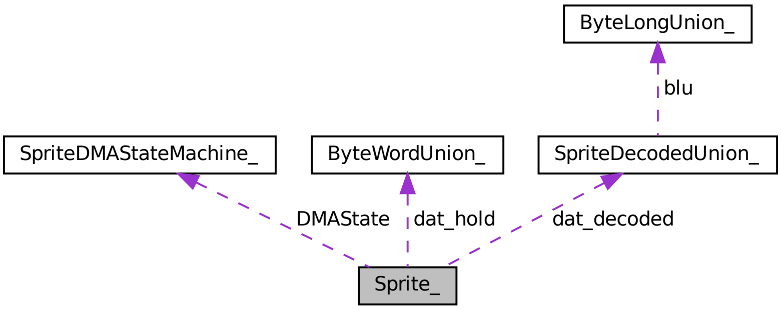
- ULO **color_i**
- UBY **color_b** [4]

The documentation for this union was generated from the following file:

- LineExactSprites.cpp

10.155 Sprite_ Struct Reference

Collaboration diagram for Sprite_:



Data Fields

- SpriteDMAStateMachine **DMAState**
- bool **armed**
- bool **attached**
- ULO **x**
- ByteWordUnion **dat_hold** [4]
- SpriteDecodedUnion **dat_decoded**
- bool **serializing**
- ULO **pixels_output**
- ULO **x_cylinder**
- UWO **dat** [2]

The documentation for this struct was generated from the following files:

- CycleExactSprites.h
- SpriteState.h

10.156 sprite_deco_ Union Reference

Data Fields

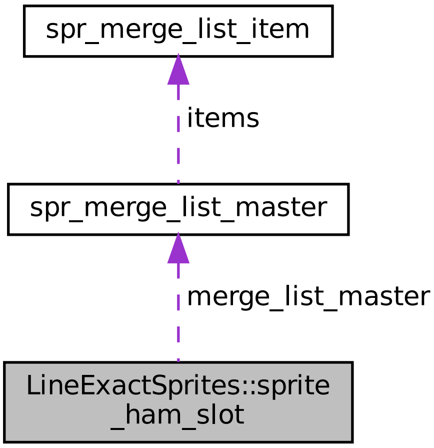
- UBY **i8** [8]
- ULO **i32** [2]

The documentation for this union was generated from the following files:

- SpriteState.c
- SpriteP2CDecoder.h

10.157 LineExactSprites::sprite_ham_slot Struct Reference

Collaboration diagram for LineExactSprites::sprite_ham_slot:



Data Fields

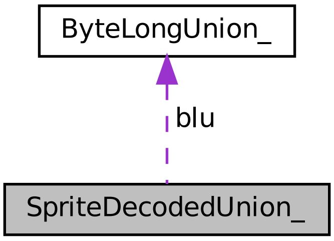
- spr_merge_list_master **merge_list_master** [8]

The documentation for this struct was generated from the following file:

- LineExactSprites.h

10.158 SpriteDecodedUnion_ Union Reference

Collaboration diagram for SpriteDecodedUnion_:



Data Fields

- UBY **barray** [16]
- ByteLongUnion **blu** [4]

The documentation for this union was generated from the following files:

- CycleExactSprites.h
- SpriteState.h

10.159 SpriteDMAStateMachine_ Struct Reference

Data Fields

- SpriteDMAStates **state**
- ULO **y_first**
- ULO **y_last**
- ULO **pt**

The documentation for this struct was generated from the following files:

- CycleExactSprites.h
- SpriteState.h

10.160 SpriteMerger Class Reference

Static Public Member Functions

- static void **MergeLores** (ULO sprite_number, UBY ∗playfield, UBY ∗sprite, ULO pixel_count)
- static void **MergeHires** (ULO sprite_number, UBY ∗playfield, UBY ∗sprite, ULO pixel_count)
- static void **MergeHam** (ULO sprite_number, UBY ∗playfield, UBY ∗ham_sprites_playfield, UBY ∗sprite, ULO pixel_count)
- static void **Initialize** ()

Static Private Attributes

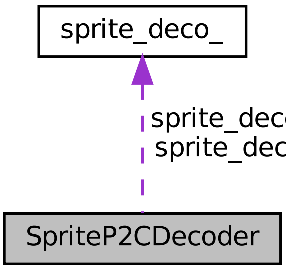
- static UBY **sprite_translate** [2][256][256]

The documentation for this class was generated from the following files:

- SpriteMerger.h
- SpriteMerger.cpp

10.161 SpriteP2CDecoder Class Reference

Collaboration diagram for SpriteP2CDecoder:



Static Public Member Functions

- static void **Decode4** (unsigned int sprite_number, ULO ∗chunky_destination, UWO data1, UWO data2)
- static void **Decode16** (ULO ∗chunky_destination, UWO data1, UWO data2, UWO data3, UWO data4)
- static void **Initialize** ()

Static Private Member Functions

- static void **P2CTablesInitialize** ()

Static Private Attributes

- static sprite_deco **sprite_deco4** [4][2][256]
- static sprite_deco **sprite_deco16** [4][256]

The documentation for this class was generated from the following files:

- SpriteP2CDecoder.h
- SpriteP2CDecoder.cpp

10.162 SpriteRegisters Class Reference

Public Member Functions

- void **InstallIOHandlers** ()
- void **ClearState** ()
- void **LoadState** (FILE ∗F)
- void **SaveState** (FILE ∗F)

Data Fields

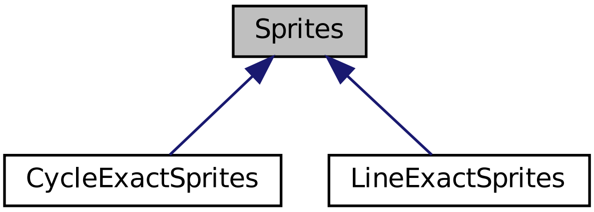
- ULO **sprpt** [8]
- UWO **sprpos** [8]
- UWO **sprctl** [8]
- UWO **sprdata** [8]
- UWO **sprdatb** [8]

The documentation for this class was generated from the following files:

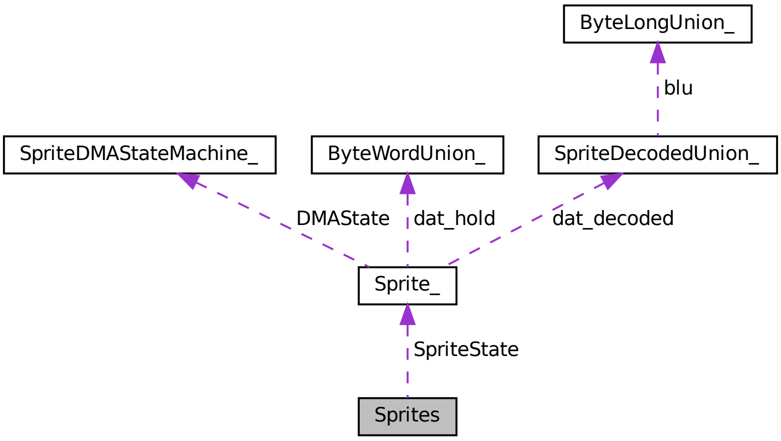
- SpriteRegisters.h
- SpriteRegisters.cpp

10.163 Sprites Class Reference

Inheritance diagram for Sprites:



Collaboration diagram for Sprites:



Public Member Functions

- void **wsprxpth** (UWO data, ULO address)
- void **wsprxptl** (UWO data, ULO address)
- void **wsprxpos** (UWO data, ULO address)
- void **wsprxctl** (UWO data, ULO address)
- void **wsprxdata** (UWO data, ULO address)
- void **wsprxdatb** (UWO data, ULO address)
- void **OutputSprite** (ULO spriteNo, ULO startCylinder, ULO cylinderCount)
- void **OutputSprites** (ULO startCylinder, ULO cylinderCount)
- void **EndOfLine** (ULO rasterY)
- void **EndOfFrame** (void)
- void **EmulationStart** (void)
- virtual void **NotifySprpthChanged** (UWO data, unsigned int sprite_number)=0
- virtual void **NotifySprptlChanged** (UWO data, unsigned int sprite_number)=0
- virtual void **NotifySprposChanged** (UWO data, unsigned int sprite_number)=0
- virtual void **NotifySprctlChanged** (UWO data, unsigned int sprite_number)=0
- virtual void **NotifySprdataChanged** (UWO data, unsigned int sprite_number)=0
- virtual void **NotifySprdatbChanged** (UWO data, unsigned int sprite_number)=0
- virtual void **EndOfLine** (ULO rasterY)=0
- virtual void **EndOfFrame** ()=0
- virtual void **HardReset** ()=0
- virtual void **EmulationStart** ()=0
- virtual void **EmulationStop** ()=0

Private Member Functions

- void **Decode4** (ULO spriteNo)
- void **Decode16** (ULO spriteNo)
- void **Arm** (ULO spriteNo)
- void **MergeLores** (ULO spriteNo, ULO source_pixel_index, ULO pixel_index, ULO pixel_count)
- void **MergeHires** (ULO spriteNo, ULO source_pixel_index, ULO pixel_index, ULO pixel_count)
- void **MergeHam** (ULO spriteNo, ULO source_pixel_index, ULO pixel_index, ULO pixel_count)
- void **Merge** (ULO spriteNo, ULO source_pixel_index, ULO pixel_index, ULO pixel_count)
- bool **InRange** (ULO spriteNo, ULO startCylinder, ULO cylinderCount)
- UWO **ReadWord** (ULO spriteNo)
- void **ReadControlWords** (ULO spriteNo)
- void **ReadDataWords** (ULO spriteNo)
- bool **IsFirstLine** (ULO spriteNo, ULO rasterY)
- bool **IsAboveFirstLine** (ULO spriteNo, ULO rasterY)
- bool **IsLastLine** (ULO spriteNo, ULO rasterY)
- bool **Is16Color** (ULO spriteNo)
- void **DMAReadControl** (ULO spriteNo, ULO rasterY)
- void **DMAReadData** (ULO spriteNo, ULO rasterY)
- void **DMAWaitingForFirstLine** (ULO spriteNo, ULO rasterY)
- void **DMAHandler** (ULO rasterY)
- void **IOHandlersInstall** (void)
- void **ClearState** (void)

Private Attributes

- Sprite **SpriteState** [8]

The documentation for this class was generated from the following files:

- SpriteState.h
- SPRITE.H
- SpriteState.c

10.164 static_mask< FB, FE > Struct Template Reference

Public Types

- enum { **value** = (0xffffffff >> FB) ^ (0xffffffff >> (FE + 1)) }

The documentation for this struct was generated from the following file:

- m68k-tester.cpp

10.165 static_mask< FB, 31 > Struct Template Reference

Public Types

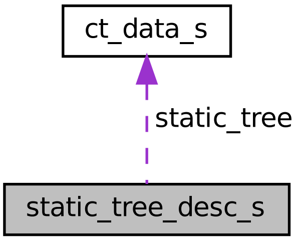
- enum { **value** = 0xffffffff >> FB }

The documentation for this struct was generated from the following file:

- m68k-tester.cpp

10.166 static_tree_desc_s Struct Reference

Collaboration diagram for static_tree_desc_s:



Data Fields

- int **dummy**
- const ct_data ∗ **static_tree**
- const intf ∗ **extra_bits**
- int **extra_base**
- int **elems**
- int **max_length**

The documentation for this struct was generated from the following files:

- deflate.c
- trees.c

10.167 stdout_state_t Struct Reference

Data Fields

- int **old_stdout**

The documentation for this struct was generated from the following file:

- m68k-tester.cpp

10.168 tagTHREADNAME_INFO Struct Reference

Data Fields

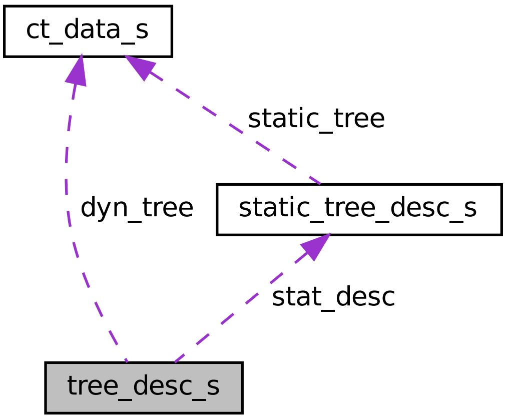
- DWORD **dwType**
- LPCSTR **szName**
- DWORD **dwThreadID**
- DWORD **dwFlags**

The documentation for this struct was generated from the following file:

- WINMAIN.C

10.169 tree_desc_s Struct Reference

Collaboration diagram for tree_desc_s:



Data Fields

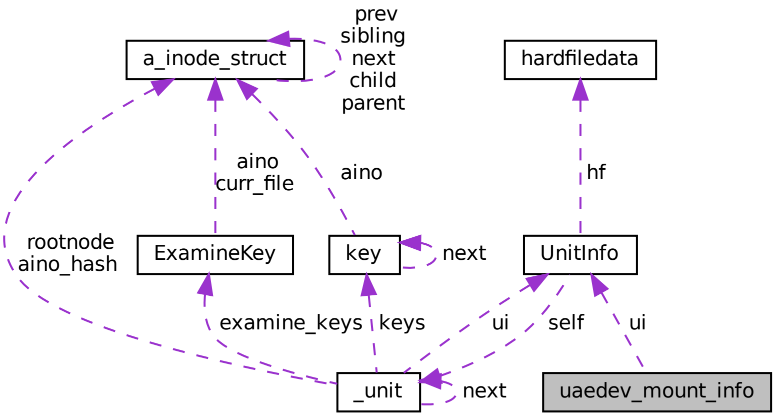
- ct_data ∗ **dyn_tree**
- int **max_code**
- static_tree_desc ∗ **stat_desc**

The documentation for this struct was generated from the following file:

- deflate.h

10.170 uaedev_mount_info Struct Reference

Collaboration diagram for uaedev_mount_info:



Data Fields

- int **num_units**
- UnitInfo **ui** [MAX_UNITS]

The documentation for this struct was generated from the following file:

- FILESYS.H

10.171 UART Class Reference

Public Member Functions

- UWO **ReadSerdatRegister** ()
- void **WriteSerdatRegister** (UWO data)
- void **WriteSerperRegister** (UWO data)
- void **NotifyInterruptRequestBitsChanged** (UWO intreq)
- void **EndOfLine** ()
- void **EndOfFrame** ()
- void **EmulationStart** ()
- void **EmulationStop** ()

Static Public Member Functions

- static void **wserper** (UWO data, ULO address)
- static void **wserdat** (UWO data, ULO address)
- static UWO **rserdat** (ULO address)

Private Member Functions

- void **InstallIOHandlers** ()
- void **ClearState** ()
- void **LoadState** (FILE ∗F)
- void **SaveState** (FILE ∗F)
- void **OpenOutputFile** ()
- void **CloseOutputFile** ()
- bool **Is8BitMode** ()
- UWO **GetBitPeriod** ()
- void **CopyReceiveShiftRegisterToBuffer** ()
- void **CopyTransmitBufferToShiftRegister** ()
- ULO **GetTransmitDoneTime** ()

Private Attributes

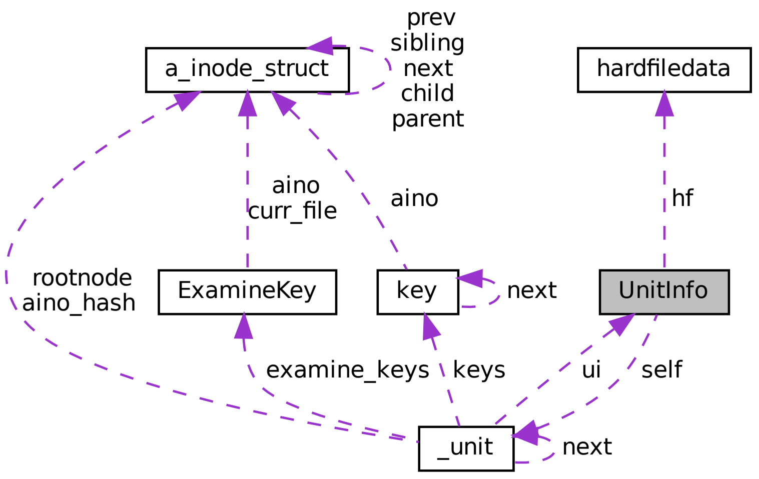
- std::string **_outputFileName**
- FILE ∗ **_outputFile**
- UWO **_serper**
- UWO **_transmitBuffer**
- UWO **_transmitShiftRegister**
- ULO **_transmitDoneTime**
- bool **_transmitBufferEmpty**
- bool **_transmitShiftRegisterEmpty**
- UWO **_receiveBuffer**
- UWO **_receiveShiftRegister**
- ULO **_receiveDoneTime**
- bool **_receiveBufferFull**
- bool **_receiveBufferOverrun**

The documentation for this class was generated from the following files:

- uart.h
- uart.cpp

10.172 UnitInfo Struct Reference

Collaboration diagram for UnitInfo:



Data Fields

- char ∗ **devname**
- uaecptr **devname_amiga**
- uaecptr **startup**
- char ∗ **volname**
- char ∗ **rootdir**
- int **readonly**
- int **devno**
- struct hardfiledata **hf**
- smp_comm_pipe ∗ **unit_pipe**
- smp_comm_pipe ∗ **back_pipe**
- uae_thread_id **tid**
- struct _unit ∗volatile **self**
- uae_sem_t **reset_sync_sem**
- int **reset_state**

The documentation for this struct was generated from the following file:

- FILESYS.H

10.173 VertexType Struct Reference

Data Fields

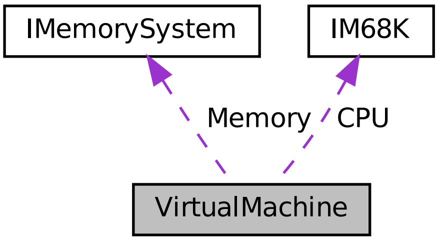
- XMFLOAT3 **position**
- XMFLOAT2 **texture**

The documentation for this struct was generated from the following file:

- GfxDrvDXGI.cpp

10.174 VirtualMachine Class Reference

Collaboration diagram for VirtualMachine:



Public Member Functions

- **VirtualMachine** (fellow::api::vm::IM68K &cpu, fellow::api::vm::IMemorySystem &memory)

Data Fields

- fellow::api::vm::IM68K & **CPU**
- fellow::api::vm::IMemorySystem & **Memory**

The documentation for this class was generated from the following file:

- VM.h

10.175 wgui_drawmode Struct Reference

Public Member Functions

- bool **operator**< (const wgui_drawmode &dm)
- **wgui_drawmode** (draw_mode ∗dm)

Data Fields

- ULO **id**
- ULO **width**
- ULO **height**
- ULO **refresh**
- ULO **colorbits**
- STR **name** [32]

The documentation for this struct was generated from the following file:

- WGUI.H

10.176 wgui_drawmodes Struct Reference

Data Fields

- ULO **numberof16bit**
- ULO **numberof24bit**
- ULO **numberof32bit**
- LON **comboxbox16bitindex**
- LON **comboxbox24bitindex**
- LON **comboxbox32bitindex**
- wgui_drawmode_list **res16bit**
- wgui_drawmode_list **res24bit**
- wgui_drawmode_list **res32bit**

The documentation for this struct was generated from the following file:

- WGUI.H

10.177 wgui_preset Struct Reference

Data Fields

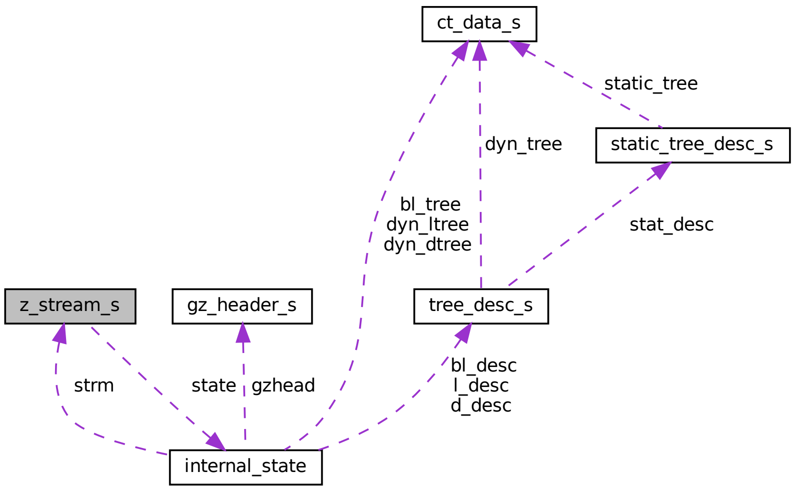
- STR **strPresetFilename** [CFG_FILENAME_LENGTH]
- STR **strPresetDescription** [CFG_FILENAME_LENGTH]

The documentation for this struct was generated from the following file:

- WGUI.H

10.178 z_stream_s Struct Reference

Collaboration diagram for z_stream_s:



Data Fields

- z_const Bytef ∗ **next_in**
- uInt **avail_in**
- uLong **total_in**
- Bytef ∗ **next_out**
- uInt **avail_out**
- uLong **total_out**
- z_const char ∗ **msg**
- struct internal_state FAR ∗ **state**
- alloc_func **zalloc**
- free_func **zfree**
- voidpf **opaque**
- int **data_type**
- uLong **adler**
- uLong **reserved**

The documentation for this struct was generated from the following file:

- zlib.h

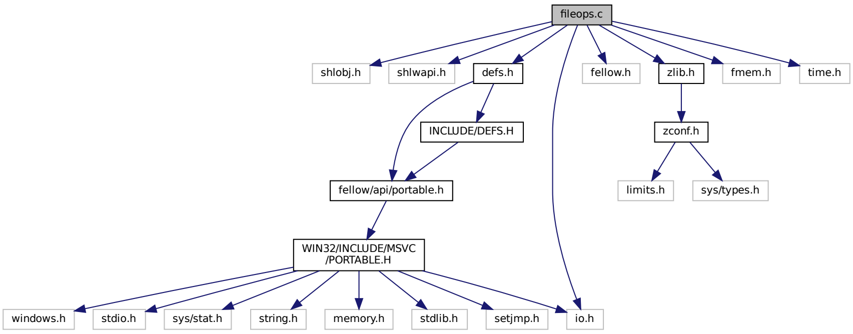
Chapter 11

File Documentation

11.1 fileops.c File Reference

```
#include <shlobj.h>
#include <shlwapi.h>
#include "defs.h"
#include "fellow.h"
#include "zlib.h"
#include "fmem.h"
#include <time.h>
#include <io.h>
```

Include dependency graph for fileops.c:



Functions

- BOOLE fileopsResolveVariables (const char ∗szPath, char ∗szNewPath)
- BOOLE fileopsGetGenericFileName (char ∗szPath, const char ∗szSubDir, const char ∗filename)
- BOOLE fileopsGetScreenshotFileName (char ∗szFilename)
- BOOLE **fileopsGetFellowLogfileName** (char ∗szPath)
- BOOLE **fileopsGetDefaultConfigFileName** (char ∗szPath)
- static BOOLE **fileopsGetWinFellowExecutablePath** (char ∗strBuffer, const DWORD lBufferSize)
- bool **fileopsGetWinFellowInstallationPath** (char ∗strBuffer, const DWORD lBufferSize)
- static bool **fileopsDirectoryExists** (const char ∗strPath)
- BOOLE **fileopsGetWinFellowPresetPath** (char ∗strBuffer, const DWORD lBufferSize)
- char ∗ **fileopsGetTemporaryFilename** (void)
- bool **fileopsGetKickstartByCRC32** (const char ∗strSearchPath, const ULO lCRC32, char ∗strDest←Filename, const ULO strDestLen)

11.1.1 Detailed Description

The fileops module contains abtract functions to generate filenames in a platform specific manner.

11.1.2 Function Documentation

11.1.2.1 fileopsGetGenericFileName()

```
BOOLE fileopsGetGenericFileName (
            char * szPath,
            const char * szSubDir,
            const char * filename )
```

build generic filename pointing to "Application Data\Roaming\WinFellow"; AmigaForever Amiga files path will be preferred over AppData when compiling a RetroPlatform specific build.

Returns

TRUE if successful, FALSE otherwise

11.1.2.2 fileopsGetScreenshotFileName()

```
BOOLE fileopsGetScreenshotFileName (
            char * szFilename )
```

generate screenshot filename (below my pictures folder)

Returns

TRUE if successful, FALSE otherwise

11.1.2.3 fileopsResolveVariables()

```
BOOLE fileopsResolveVariables (
            const char * szPath,
            char * szNewPath )
```

resolve environment variables in file/folder names.

Parameters

in	*szPath*	path name to resolve
out	*szNewPath*	path name with resolved variables

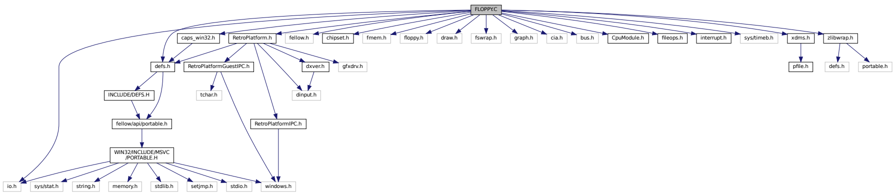
Returns

TRUE, if variable was successfully resolved, FALSE otherwise.

11.2 FLOPPY.C File Reference

```
#include <io.h>
#include "defs.h"
#include "fellow.h"
#include "chipset.h"
#include "fmem.h"
#include "floppy.h"
#include "draw.h"
#include "fswrap.h"
#include "graph.h"
#include "cia.h"
#include "bus.h"
#include "CpuModule.h"
#include "fileops.h"
#include "interrupt.h"
#include <sys/timeb.h>
#include "xdms.h"
#include "zlibwrap.h"
#include "caps_win32.h"
#include "RetroPlatform.h"
```

Include dependency graph for FLOPPY.C:



Macros

- #define **MFM_FILLB** 0xaa
- #define **MFM_FILLL** 0xaaaaaaaa
- #define **MFM_MASK** 0x55555555
- #define **FLOPPY_INSERTED_DELAY** 150
- #define **FLOPPY_GAP_BYTES** 720
- #define **FLOPPY_FAST_WORDS** 32

Functions

- UWO **radcon** (ULO address)
- void **wadcon** (UWO data, ULO address)
- UWO **rdskbytr** (ULO address)
- void **wdskpth** (UWO data, ULO address)
- void **wdskptl** (UWO data, ULO address)
- void **floppyClearDMAState** ()

- void **wdsklen** (UWO data, ULO address)
- void **wdsksync** (UWO data, ULO address)
- LON **floppySelectedGet** (void)
- void **floppySelectedSet** (ULO selbits)
- BOOLE **floppyIsTrack0** (ULO drive)
- BOOLE **floppyIsWriteProtected** (ULO drive)
- BOOLE **floppyIsReady** (ULO drive)
- BOOLE **floppyIsChanged** (ULO drive)
- void **floppyMotorSet** (ULO drive, BOOLE mtr)
- void **floppySideSet** (BOOLE s)
- void **floppyDirSet** (BOOLE dr)
- void floppyStepSet (BOOLE stp)
- void **floppySectorMfmEncode** (ULO tra, ULO sec, UBY ∗src, UBY ∗dest, ULO sync)
- void **floppyGapMfmEncode** (UBY ∗dst)
- ULO **floppySectorMfmDecode** (UBY ∗src, UBY ∗dst, ULO track)
- BOOLE **floppySectorSave** (ULO drive, ULO track, UBY ∗mfmsrc)
- void **floppyTrackMfmEncode** (ULO track, UBY ∗src, UBY ∗dst, ULO sync)
- void **floppyTrackLoad** (ULO drive, ULO track)
- void **floppyError** (ULO drive, ULO errorID)
- static void floppyWriteDiskDate (UBY ∗strBuffer)
- static void floppyWriteDiskChecksum (const UBY ∗strBuffer, UBY ∗strChecksum)
- bool **floppyValidateAmigaDOSVolumeName** (const STR ∗strVolumeName)
- static void **floppyWriteDiskBootblock** (UBY ∗strCylinderContent, bool bFFS, bool bBootable)
- static void **floppyWriteDiskRootBlock** (UBY ∗strCylinderContent, ULO lBlockIndex, const UBY ∗strVolumeLabel)
- bool **floppyImageADFCreate** (STR ∗strImageFilename, STR ∗strVolumeLabel, bool bFormat, bool bBootable, bool bFFS)
- BOOLE **floppyImageCompressedBZipPrepare** (STR ∗diskname, ULO drive)
- BOOLE **floppyImageCompressedDMSPrepare** (STR ∗diskname, ULO drive)
- BOOLE **floppyImageCompressedGZipPrepare** (STR ∗diskname, ULO drive)
- void **floppyImageCompressedRemove** (ULO drive)
- BOOLE **floppyImageCompressedPrepare** (STR ∗diskname, ULO drive)
- void **floppyImageRemove** (ULO drive)
- void **floppyImagePrepare** (STR ∗diskname, ULO drive)
- ULO **floppyImageGeometryCheck** (fs_navig_point ∗fsnp, ULO drive)
- void **floppyImageNormalLoad** (ULO drive)
- void **floppyImageExtendedLoad** (ULO drive)
- void **floppyImageIPFLoad** (ULO drive)
- void floppySetDiskImage (ULO drive, STR ∗diskname)
- void **floppySetEnabled** (ULO drive, BOOLE enabled)
- void floppySetReadOnly (ULO drive, BOOLE readonly)
- void **floppySetFastDMA** (BOOLE fastDMA)
- void **floppyDriveTableInit** (void)
- void **floppyDriveTableReset** (void)
- void **floppyMfmDataFree** (void)
- void **floppyTimeBufDataFree** (void)
- void **floppyIOHandlersInstall** (void)
- void **floppyIORegistersClear** (void)
- BOOLE **floppyDMAReadStarted** (void)
- BOOLE **floppyDMAWriteStarted** (void)
- BOOLE **floppyDMAChannelOn** (void)
- BOOLE **floppyHasIndex** (ULO sel_drv)
- ULO **floppyGetLinearTrack** (ULO sel_drv)
- BOOLE **floppyIsSpinning** (ULO sel_drv)
- void **floppyDMAReadInit** (ULO drive)

- ULO **floppyFindNextSync** (ULO pos, LON length)
- void **floppyDMAWriteInit** (LON drive)
- void **floppyDMAStart** (void)
- void **floppyDMAWrite** (void)
- BOOLE **floppyCheckSync** (UWO word_under_head)
- void **floppyReadWord** (UWO word_under_head, BOOLE found_sync)
- UWO **floppyGetByteUnderHead** (ULO sel_drv, ULO track)
- void **floppyNextByte** (ULO sel_drv, ULO track)
- void **floppyEndOfLine** (void)
- void **floppyHardReset** (void)
- void **floppyEmulationStart** (void)
- void **floppyEmulationStop** (void)
- void **floppyStartup** (void)
- void **floppyShutdown** (void)

Variables

- floppyinfostruct **floppy** [4]
- BOOLE **floppy_fast**
- UBY **tmptrack** [20 ∗1024 ∗11]
- floppyDMAinfostruct **floppy_DMA**
- BOOLE **floppy_DMA_started**
- BOOLE **floppy_DMA_read**
- BOOLE **floppy_has_sync**
- ULO **dsklen**
- ULO **dsksync**
- ULO **dskpt**
- ULO **dskbytr**
- UWO **adcon**
- ULO **diskDMAen**
- UWO **dskbyt_tmp** = 0
- BOOLE **dskbyt1_read** = FALSE
- BOOLE **dskbyt2_read** = FALSE
- static UBY **floppyBootBlockOFS** [ ]
- static UBY **floppyBootBlockFFS** [ ]
- UWO **prev_byte_under_head** = 0

11.2.1 Detailed Description

The floppy module handles floppy disc drive emulation. It supports the use of .adf files, and is able to handle gzip (via embedded zlib code) and xdms (also embedded) compressed disc images.

It contains experimental support for .ipf files originating from the C.A.P.S. project `Software Preservation Society`). CAPS ∗ support is not yet fully functional, because timings are not emulated correctly to support copy-protected ("flakey") images.

CAPS support is only available for the 32 bit version, since no 64 bit version of the library is available. CAPS support is only enabled, when the preprocessor definition FELLOW_SUPPORT_CAPS is set - this should be disabled for 64 bit builds.

Todo CAPS has been renamed to SPS, and a 64 bit version is available; update to a current version

enhance timing for flakey image support

11.2.2 Function Documentation

11.2.2.1 floppySetDiskImage()

```
void floppySetDiskImage (
             ULO drive,
             STR * diskname )
```

Insert an image into a floppy drive

11.2.2.2 floppySetReadOnly()

```
void floppySetReadOnly (
             ULO drive,
             BOOLE readonly )
```

Set read-only flag for a drive.

11.2.2.3 floppyStepSet()

```
void floppyStepSet (
             BOOLE stp )
```

Move the floppy head to a given position (step).

11.2.2.4 floppyWriteDiskChecksum()

```
static void floppyWriteDiskChecksum (
             const UBY * strBuffer,
             UBY * strChecksum )  [static]
```

Write the checksum into the floppy disk buffer.

11.2.2.5 floppyWriteDiskDate()

```
static void floppyWriteDiskDate (
             UBY * strBuffer )  [static]
```

Write the current date/time into the floppy disk buffer.

11.2.3 Variable Documentation

11.2.3.1 floppyBootBlockFFS

```
UBY floppyBootBlockFFS[]  [static]
```

Initial value:

```
={
  0x44, 0x4F, 0x53, 0x01, 0xE3, 0x3D, 0x0E, 0x72, 0x00, 0x00, 0x03, 0x70, 0x43, 0xFA, 0x00, 0x3E,
  0x70, 0x25, 0x4E, 0xAE, 0xFD, 0xD8, 0x4A, 0x80, 0x67, 0x0C, 0x22, 0x40, 0x08, 0xE9, 0x00, 0x06,
  0x00, 0x22, 0x4E, 0xAE, 0xFE, 0x62, 0x43, 0xFA, 0x00, 0x18, 0x4E, 0xAE, 0xFF, 0xA0, 0x4A, 0x80,
  0x67, 0x0A, 0x20, 0x40, 0x20, 0x68, 0x00, 0x16, 0x70, 0x00, 0x4E, 0x75, 0x70, 0xFF, 0x4E, 0x75,
  0x64, 0x6F, 0x73, 0x2E, 0x6C, 0x69, 0x62, 0x72, 0x61, 0x72, 0x79, 0x00, 0x65, 0x78, 0x70, 0x61,
  0x6E, 0x73, 0x69, 0x6F, 0x6E, 0x2E, 0x6C, 0x69, 0x62, 0x72, 0x61, 0x72, 0x79, 0x00, 0x00, 0x00,
}
```

11.2.3.2 floppyBootBlockOFS

```
UBY floppyBootBlockOFS[]  [static]
```

Initial value:

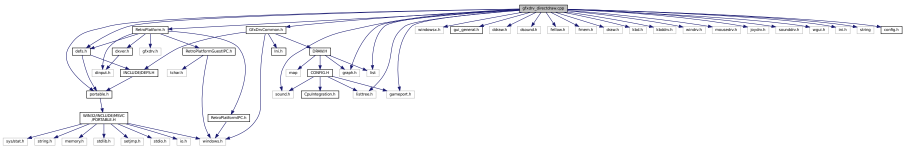
```
={
  0x44, 0x4f, 0x53, 0x00, 0xc0, 0x20, 0x0f, 0x19, 0x00, 0x00, 0x03, 0x70, 0x43, 0xfa, 0x00, 0x18,
  0x4e, 0xae, 0xff, 0xa0, 0x4a, 0x80, 0x67, 0x0a, 0x20, 0x40, 0x20, 0x68, 0x00, 0x16, 0x70, 0x00,
  0x4e, 0x75, 0x70, 0xff, 0x60, 0xfa, 0x64, 0x6f, 0x73, 0x2e, 0x6c, 0x69, 0x62, 0x72, 0x61, 0x72,
  0x79
}
```

11.3 gfxdrv_directdraw.cpp File Reference

```
#include "portable.h"
#include <windowsx.h>
#include "gui_general.h"
#include <ddraw.h>
#include <dsound.h>
#include "defs.h"
#include "fellow.h"
#include "fmem.h"
#include "sound.h"
#include "graph.h"
#include "draw.h"
#include "kbd.h"
#include "kbddrv.h"
#include "listtree.h"
#include "windrv.h"
#include "gameport.h"
#include "mousedrv.h"
#include "joydrv.h"
#include "sounddrv.h"
#include "wgui.h"
#include "ini.h"
#include <list>
#include <string>
#include "GfxDrvCommon.h"
```

```
#include "RetroPlatform.h"
#include "config.h"
```

Include dependency graph for gfxdrv_directdraw.cpp:



Data Structures

- struct gfx_drv_ddraw_fullscreen_mode
- struct gfx_drv_ddraw_device

Functions

- const STR ∗ **gfxDrvDDrawErrorString** (HRESULT hResult)
- void **gfxDrvDDrawPrintPixelFlags** (DWORD flags, STR ∗s)
- void **gfxDrvDDrawFailure** (const STR ∗header, HRESULT err)
- void **gfxDrvDDrawFindWindowClientRect** (gfx_drv_ddraw_device ∗ddraw_device)
- void **gfxDrvDDrawDrawTargetSurfaceSelect** (gfx_drv_ddraw_device ∗ddraw_device, LPDIRECTDRAW↩ SURFACE ∗lpDDS, LPDDSURFACEDESC ∗lpDDSD)
- void **gfxDrvDDrawBlitTargetSurfaceSelect** (gfx_drv_ddraw_device ∗ddraw_device, LPDIRECTDRAWS↩ URFACE ∗lpDDS)
- void **gfxDrvDDrawClipperRelease** (gfx_drv_ddraw_device ∗ddraw_device)
- bool **gfxDrvDDrawClipperInitialize** (gfx_drv_ddraw_device ∗ddraw_device)
- BOOL WINAPI **gfxDrvDDrawDeviceEnumerate** (GUID FAR ∗lpGUID, LPSTR lpDriverDescription, LPSTR lpDriverName, LPVOID lpContext)
- void **gfxDrvDDrawDeviceInformationDump** ()
- bool **gfxDrvDDrawDeviceInformationInitialize** ()
- void **gfxDrvDDrawDeviceInformationRelease** ()
- bool **gfxDrvDDraw1ObjectInitialize** (gfx_drv_ddraw_device ∗ddraw_device)
- bool **gfxDrvDDraw2ObjectInitialize** (gfx_drv_ddraw_device ∗ddraw_device)
- bool **gfxDrvDDraw1ObjectRelease** (gfx_drv_ddraw_device ∗ddraw_device)
- bool **gfxDrvDDraw2ObjectRelease** (gfx_drv_ddraw_device ∗ddraw_device)
- bool **gfxDrvDDrawObjectInitialize** (gfx_drv_ddraw_device ∗ddraw_device)
- gfx_drv_ddraw_fullscreen_mode ∗ **gfxDrvDDrawFindFullScreenMode** (gfx_drv_ddraw_device ∗ddraw_↩ device, ULO width, ULO height, ULO depth)
- void **gfxDrvDDrawRegisterFullScreenModeInformation** (gfx_drv_ddraw_device ∗ddraw_device)
- ULO **gfxDrvRGBMaskPos** (ULO mask)
- ULO **gfxDrvRGBMaskSize** (ULO mask)
- void **gfxDrvDDrawLogFullScreenModeInformation** (gfx_drv_ddraw_device ∗ddraw_device)
- gfx_drv_ddraw_fullscreen_mode ∗ **gfxDrvDDrawNewFullScreenMode** (ULO width, ULO height, ULO depth, ULO refresh, ULO redpos, ULO redsize, ULO greenpos, ULO greensize, ULO bluepos, ULO bluesize)
- HRESULT WINAPI **gfxDrvDDrawEnumerateFullScreenMode** (LPDDSURFACEDESC lpDDSurfaceDesc, LPVOID lpContext)
- bool **gfxDrvDDrawInitializeFullScreenModeInformation** (gfx_drv_ddraw_device ∗ddraw_device)
- void **gfxDrvDDrawReleaseFullScreenModeInformation** (gfx_drv_ddraw_device ∗ddraw_device)
- bool **gfxDrvDDrawSetCooperativeLevelNormal** (gfx_drv_ddraw_device ∗ddraw_device)
- bool **gfxDrvDDrawSetCooperativeLevelExclusive** (gfx_drv_ddraw_device ∗ddraw_device)

- bool **gfxDrvDDrawSetCooperativeLevel** (gfx_drv_ddraw_device ∗ddraw_device)
- void **gfxDrvDDrawSurfaceClear** (gfx_drv_ddraw_device ∗ddraw_device, LPDIRECTDRAWSURFACE surface, RECT ∗dstrect=nullptr)
- HRESULT **gfxDrvDDrawSurfaceRestore** (gfx_drv_ddraw_device ∗ddraw_device, LPDIRECTDRAWSU↩ RFACE surface)
- void **gfxDrvDDrawCalculateDestinationRectangle** (ULO output_width, ULO output_height, gfx_drv_↩ ddraw_device ∗ddraw_device, RECT &dstwin)
- void **gfxDrvDDrawSurfaceBlit** (gfx_drv_ddraw_device ∗ddraw_device)
- void **gfxDrvDDrawSurfacesRelease** (gfx_drv_ddraw_device ∗ddraw_device)
- const char ∗ **gfxDrvDDrawVideomemLocationStr** (ULO pass)
- ULO **gfxDrvDDrawVideomemLocationFlags** (ULO pass)
- bool **gfxDrvDDrawCreateSecondaryOffscreenSurface** (gfx_drv_ddraw_device ∗ddraw_device)
- ULO **gfxDrvDDrawSurfacesInitialize** (gfx_drv_ddraw_device ∗ddraw_device)
- void **gfxDrvDDrawClearWindowBorders** (gfx_drv_ddraw_device ∗ddraw_device)
- UBY ∗ **gfxDrvDDrawSurfaceLock** (gfx_drv_ddraw_device ∗ddraw_device, ULO ∗pitch)
- void **gfxDrvDDrawSurfaceUnlock** (gfx_drv_ddraw_device ∗ddraw_device)
- void **gfxDrvDDrawFlip** ()
- unsigned int **gfxDrvDDrawSetMode** (gfx_drv_ddraw_device ∗ddraw_device)
- void **gfxDrvDDrawGetBufferInformation** (draw_buffer_information ∗buffer_information)
- bool **gfxDrvDDrawInitialize** ()
- void **gfxDrvDDrawRelease** ()
- void **gfxDrvDDrawClearCurrentBuffer** ()
- UBY ∗ **gfxDrvDDrawValidateBufferPointer** ()
- void **gfxDrvDDrawInvalidateBufferPointer** ()
- void **gfxDrvDDrawSizeChanged** (unsigned int width, unsigned int height)
- void **gfxDrvDDrawPositionChanged** ()
- void **gfxDrvDDrawSetMode** (draw_mode ∗dm, bool windowed)
- bool gfxDrvDDrawEmulationStart (ULO maxbuffercount)
- unsigned int gfxDrvDDrawEmulationStartPost ()
- void **gfxDrvDDrawEmulationStop** ()
- bool **gfxDrvDDrawStartup** ()
- void **gfxDrvDDrawShutdown** ()
- bool **gfxDrvDDrawSaveScreenshotFromDCArea** (HDC hDC, DWORD x, DWORD y, DWORD width, D↩ WORD height, ULO lDisplayScale, DWORD bits, const STR ∗filename)
- static bool **gfxDrvDDrawSaveScreenshotFromSurfaceArea** (LPDIRECTDRAWSURFACE surface, DW↩ ORD x, DWORD y, DWORD width, DWORD height, ULO lDisplayScale, const STR ∗filename)
- bool **gfxDrvDDrawSaveScreenshot** (const bool bTakeFilteredScreenshot, const STR ∗filename)

Variables

- gfx_drv_ddraw_device ∗ **gfx_drv_ddraw_device_current**
- felist ∗ **gfx_drv_ddraw_devices**
- bool **gfx_drv_ddraw_initialized**
- bool **gfx_drv_ddraw_clear_borders**
- ULO **gfx_drv_output_width**
- ULO **gfx_drv_output_height**

11.3.1 Detailed Description

Graphics device module

Framebuffer modes of operation:

1. Using the primary buffer non-clipped, with possible back-buffers. This applies to fullscreen mode with a framepointer.
2. Using a secondary buffer to render and then applying the blitter to the primary buffer. The primary buffer can be clipped (window) or non-clipped (fullscreen). In this mode there are no backbuffers.

blitmode is used:

1. When running on the desktop in a window.
2. In fullscreen mode with a primary surface that can not supply a framebuffer pointer.

Windows: Two types of windows:

1. Normal desktop window for desktop operation
2. Full-screen window for fullscreen mode

Windows are created and destroyed on emulation start and stop.

Buffers: Buffers are created when emulation starts and destroyed when emulation stops. (Recreated when lost also)

if blitmode, create single primary buffer and a secondary buffer for actual rendering in system memory. if windowed also add a clipper to the primary buffer

else create a primary buffer (with backbuffers)

11.3.2 Function Documentation

11.3.2.1 gfxDrvDDrawEmulationStart()

```
bool gfxDrvDDrawEmulationStart (
            ULO maxbuffercount )
```

Emulation is starting.

Called on emulation start. Subtlety: In exclusive mode, the window that is attached to the device appears to become activated, even if it is not shown at the time. The WM_ACTIVATE message triggers DirectInput acquisition, which means that the DirectInput object needs to have been created at that time. Unfortunately, the window must be created as well in order to attach DI objects to it. So we create window, create DI objects in between and then do the rest of the gfx init. That is why gfxDrvEmulationStart is split in two.

11.3.2.2 gfxDrvDDrawEmulationStartPost()

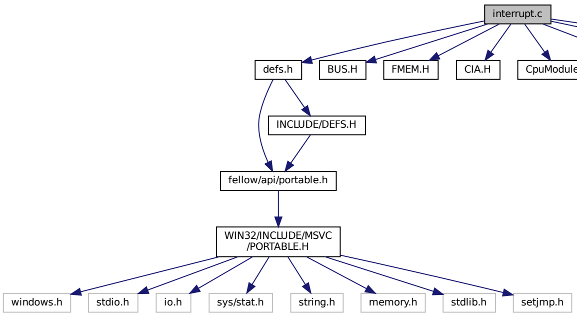
```
unsigned int gfxDrvDDrawEmulationStartPost ( )
```

Emulation is starting, post

11.4 interrupt.c File Reference

```
#include "defs.h"
#include "BUS.H"
#include "FMEM.H"
#include "CIA.H"
#include "CpuModule.h"
#include "CpuIntegration.h"
#include "UART.h"
#include "fileops.h"
```

Include dependency graph for interrupt.c:



Functions

- STR ∗ **interruptGetInterruptName** (ULO interrupt_number)
- void **interruptSetPendingChipInterruptNumber** (ULO pending_chip_interrupt_number)
- ULO **interruptGetPendingChipInterruptNumber** (void)
- void **interruptSetPendingCpuLevel** (ULO pending_cpu_level)
- ULO **interruptGetPendingCpuLevel** (void)
- unsigned int **interruptGetScheduleLatency** (void)
- unsigned int **interruptGetCpuLevel** (int interrupt_number)
- bool **interruptIsPending** (int interrupt_number, unsigned int pending_interrupts)
- bool **interruptMasterSwitchIsEnabled** (void)
- bool **interruptHasSetModeBit** (UWO interrupt_bitmask)
- UWO **interruptGetPendingBitMask** (void)
- UWO **interruptSetBits** (UWO original, UWO set_bitmask)
- UWO **interruptClearBits** (UWO original, UWO clear_bitmask)
- BOOLE **interruptIsRequested** (UWO bitmask)
- void **interruptHandleEvent** (void)
- BOOLE **cpuGetRaiseInterrupt** (void)
- void **interruptRaisePending** (void)
- UWO **rintreqr** (ULO address)

- void **wintreq** (UWO data, ULO address)
- UWO **rintenar** (ULO address)
- void **wintena** (UWO data, ULO address)
- void **interruptClearInternalState** (void)
- void **interruptIoHandlersInstall** (void)
- void **interruptSoftReset** (void)
- void **interruptHardReset** (void)
- void **interruptEmulationStart** (void)
- void **interruptEmulationStop** (void)
- void **interruptStartup** (void)
- void **interruptShutdown** (void)

Variables

- UWO **intena**
- UWO **intreq**
- ULO **interrupt_pending_cpu_level**
- ULO **interrupt_pending_chip_interrupt_number**
- static unsigned int **interrupt_cpu_level** [16] = {1,1,1,2, 3,3,3,4, 4,4,4,5, 5,6,6,7}

11.4.1 Detailed Description

Chipset side of interrupt control

The process of servicing interrupts is asynchronous in several steps

Case 1: Chipset requests an interrupt, or program sets intreq or intena.

1. interruptRaisePending() is called to evaluate the current requested and enabled interrupts.
2. If one is found, to emulate the chipset latency before actually sending the desired interrupt level to the CPU, the interrupt event is used (bus.c), scheduled to fire some cycles from now.
3. The interrupt event fires, calls interruptHandleEvent() which will set the new interrupt level in the cpu using cpuSetIrqLevel(). The rest in the hands of the cpu module.
4. cpuSetIrqLevel() will set an internal flag, record the new interrupt level, and unstop the CPU if needed. CPU state is not changed here.
5. The next time cpuExecuteIntruction runs, it will switch to the new interrupt level and make the necessary state changes.

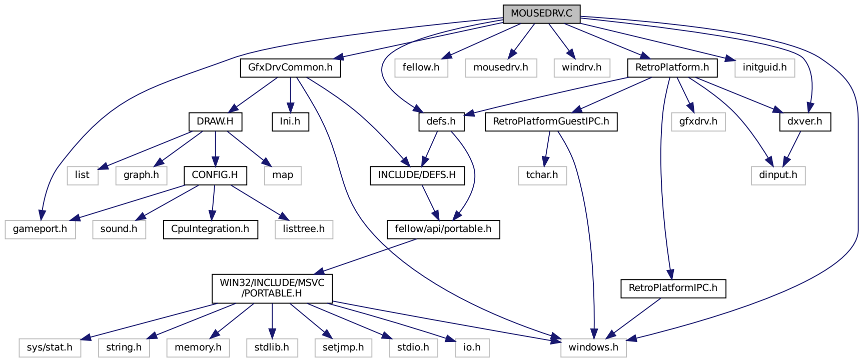
Case 2: CPU lowers the interrupt level (RTE, write to SR etc).

1. The CPU module calls the checkPendingInterrupt hook to force us to re-evaluate the interrupt sources. This hook points to interruptRaisePending(). Continues as Case 1.

11.5 MOUSEDRV.C File Reference

```
#include "defs.h"
#include <windows.h>
#include "gameport.h"
#include "fellow.h"
#include "mousedrv.h"
#include "windrv.h"
#include "GfxDrvCommon.h"
#include <initguid.h>
#include "dxver.h"
#include "RetroPlatform.h"
```

Include dependency graph for MOUSEDRV.C:



Macros

- #define **DINPUT_BUFFERSIZE** 16
- #define **INITDIPROP**(diprp, obj, how)

Functions

- STR ∗ **mouseDrvDInputErrorString** (HRESULT hResult)
- void **mouseDrvDInputFailure** (STR ∗header, HRESULT err)
- void **mouseDrvDInputAcquire** (void)
- void **mouseDrvDInputRelease** (void)
- BOOL FAR PASCAL **GetMouseInfo** (LPCDIDEVICEINSTANCE pdinst, LPVOID pvRef)
- BOOLE **mouseDrvDInputInitialize** (void)
- void **mouseDrvStateHasChanged** (BOOLE active)
- void **mouseDrvToggleFocus** (void)
- void mouseDrvSetFocus (const BOOLE bNewFocus, const BOOLE bRequestedByRPHost)
- void **mouseDrvMovementHandler** (void)
- BOOLE **mouseDrvGetFocus** (void)
- void **mouseDrvHardReset** (void)
- BOOLE **mouseDrvEmulationStart** (void)
- void **mouseDrvEmulationStop** (void)
- void **mouseDrvStartup** (void)
- void **mouseDrvShutdown** (void)

Variables

- LPDIRECTINPUT **mouse_drv_lpDI** = NULL
- LPDIRECTINPUTDEVICE **mouse_drv_lpDID** = NULL
- HANDLE **mouse_drv_DIevent** = NULL
- BOOLE **mouse_drv_focus**
- BOOLE **mouse_drv_active**
- BOOLE **mouse_drv_in_use**
- BOOLE **mouse_drv_initialization_failed**
- static BOOLE **bLeftButton**
- static BOOLE **bRightButton**
- int **num_mouse_attached** = 0

11.5.1 Detailed Description

Mouse driver for Windows

11.5.2 Macro Definition Documentation

11.5.2.1 INITDIPROP

```
#define INITDIPROP(
              diprp,
              obj,
              how )
```

Value:

```
{ diprp.diph.dwSize = sizeof( diprp ); \
    diprp.diph.dwHeaderSize = sizeof( diprp.diph ); \
    diprp.diph.dwObj        = obj; \
    diprp.diph.dwHow        = how; }
```

11.5.3 Function Documentation

11.5.3.1 mouseDrvSetFocus()

```
void mouseDrvSetFocus (
             const BOOLE bNewFocus,
             const BOOLE bRequestedByRPHost )
```

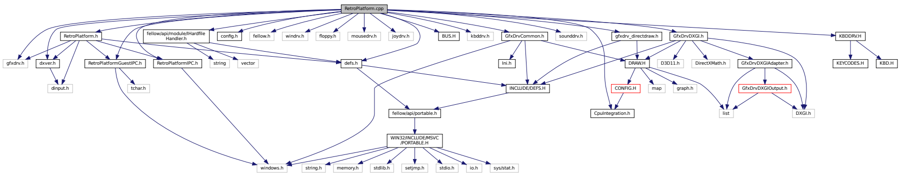
Mouse set focus

Used by the RetroPlatform module to control the mouse focus state; the player is notified of the state change only if a change was not requested by the player itself.

11.6 RetroPlatform.cpp File Reference

```
#include "defs.h"
#include "RetroPlatform.h"
#include "RetroPlatformGuestIPC.h"
#include "RetroPlatformIPC.h"
#include "config.h"
#include "fellow.h"
#include "windrv.h"
#include "floppy.h"
#include "gfxdrv.h"
#include "mousedrv.h"
#include "joydrv.h"
#include "CpuIntegration.h"
#include "BUS.H"
#include "kbddrv.h"
#include "fellow/api/module/IHardfileHandler.h"
#include "dxver.h"
#include "sounddrv.h"
#include "GfxDrvCommon.h"
#include "gfxdrv_directdraw.h"
#include "GfxDrvDXGI.h"
#include "KBDDRV.H"
```

Include dependency graph for RetroPlatform.cpp:



Functions

- BOOL RetroPlatformHandleIncomingGuestEvent (STR ∗strCurrentEvent)
- BOOL **RetroPlatformHandleIncomingGuestEventMessageParser** (STR ∗strEventMessage)
- BOOL **RetroPlatformHandleIncomingGuestEventMessage** (wchar_t ∗wcsEventMessage)
- BOOL **RetroPlatformHandleIncomingDeviceActivity** (WPARAM wParam, LPARAM lParam)
- LRESULT CALLBACK **RetroPlatformHostMessageFunction** (UINT uMessage, WPARAM wParam, LPARAM lParam, LPCVOID pData, DWORD dwDataSize, LPARAM lMsgFunctionParam)
- BOOL FAR PASCAL **RetroPlatformEnumerateJoystick** (LPCDIDEVICEINSTANCE pdinst, LPVOID pvRef)

Variables

- RetroPlatform RP

  needed for DirectSound volume control

11.6.1 Detailed Description

Cloanto RetroPlatform GUI integration.

This module contains RetroPlatform specific functionality to register as RetroPlatform guest and interact with the host (player). It imitates the full Windows GUI module, implementing the same functionality, but supported via the RetroPlatform player as main GUI. WinFellow's own GUI is not shown, the emulator operates in a headless mode. The configuration is received as a command line parameter, all control events (start, shutdown, reset, ...) are sent via IPC.

Important Note: The Cloanto modules make heavy use of unicode strings. As WinFellow uses ANSI strings, conversion is usually required (using, for example, wsctombs and mbstowcs).

When looking at an RP9 package, the RetroPlatform WinFellow plug-in has a list of criteria to decide if WinFellow is a compatible emulator that is offered as choice. It verifies that

- a valid model-specific INI file exists for the configured Amiga model
- no extended ADF files are used (file size = 901.120 for all ADFs)

The plug-in will block the start of WinFellow on a number of criteria:

- if filesystems are used
- hardfiles using a non-standard geometry, or RDB hardfiles

The RetroPlatform WinFellow plug-in will start WinFellow with the following command-line arguments:

-rphost: ID of the RetroPlatform host, used to initiate IPC communication.

-datapath: the path where WinFellow specific runtime information should be stored.

-f: The -f parameter provides an initial configuration, that is created in the following order:

1. the WinFellow plug-in's shared.ini is applied
2. a model-specific INI is applied on top of that
3. the WinFellow plug-in overrides a number of things:
    - if "No interlace" is checked in the compatibility settings, fellow.gfx_deinterlace is set to no, otherwise to yes
    - if "Turbo CPU" is checked, cpu_speed is set to 0
    - if the user has enabled "Always speed up drives where possible", "Turbo floppy" is set to yes in the RP9, and "Always use more accurate (slower) emulation" in the Option dialog is NOT set, fellow.floppy_fast_dma is set to yes, otherwise to no
    - gfx_width, gfx_height, gfx_offset_left and gfx_offset_top are added into the configuration depending on the settings of the RP9; the numbers assume the maximum pixel density (horizontal values super hires, vertical values interlace), so depending on the mode displayed, conversion is required; the clipping offsets need to be adjusted (384 to the top, 52 to the left)
4. the WinFellow plug-in's override.ini is applied on top of that, to apply any settings that always must be active

-rpescapekey: the DirectInput keycode of the escape key

-rpescapeholdtime: the time in milliseconds after which the escape key should actually escape

-rpscreenmode: the initial screenmode the guest should activate at startup (e.g. 1X, 2X, ...). It is the numerical equivalent of the RP_SCREENMODE_∗ flags (see RetroPlatformIPC.h).

11.6.2 Function Documentation

11.6.2.1 RetroPlatformHandleIncomingGuestEvent()

```
BOOL RetroPlatformHandleIncomingGuestEvent (
             STR * strCurrentEvent )
```

event handler function for events that are sent to WinFellow from Amiga Forever handles multiple incoming events like keyboard or joystick input events that are queued within the event message returns TRUE if successful, FALSE otherwise (for instance if an unrecogized event is encountered)

11.6.3 Variable Documentation

11.6.3.1 RP

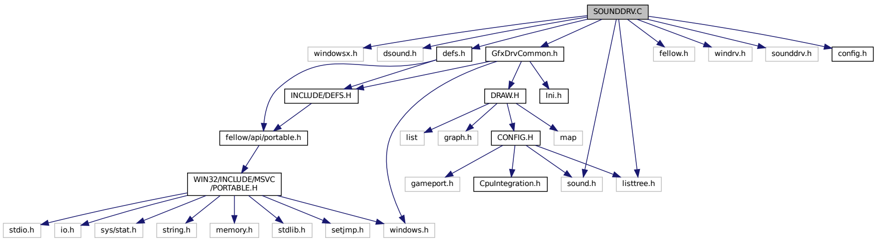
```
RetroPlatform RP
```

needed for DirectInput based joystick detection code

11.7 SOUNDDRV.C File Reference

```
#include <windowsx.h>
#include <dsound.h>
#include "defs.h"
#include "fellow.h"
#include "sound.h"
#include "listtree.h"
#include "windrv.h"
#include "sounddrv.h"
#include "config.h"
#include "GfxDrvCommon.h"
```

Include dependency graph for SOUNDDRV.C:



Data Structures

- struct sound_drv_dsound_mode
- struct sound_drv_dsound_device

Functions

- STR ∗ **soundDrvDSoundErrorString** (HRESULT hResult)
- void **soundDrvPollBufferPosition** (void)
- void CALLBACK **timercb** (UINT uID, UINT uMsg, DWORD_PTR dwUser, DWORD_PTR dw1, DWORD_↩ PTR dw2)
- void **soundDrvDSoundFailure** (STR ∗header, HRESULT err)
- void **soundDrvDSoundRelease** (void)
- bool **soundDrvDSoundInitialize** (void)
- void **soundDrvAddMode** (sound_drv_dsound_device ∗dsound_device, bool stereo, bool bits16, ULO rate)
- sound_drv_dsound_mode ∗ **soundDrvFindMode** (sound_drv_dsound_device ∗dsound_device, bool stereo, bool bits16, ULO rate)
- void **soundDrvDSoundModeInformationRelease** (sound_drv_dsound_device ∗dsound_device)
- void **soundDrvYesNoLog** (STR ∗intro, bool pred)
- bool **soundDrvDSoundModeInformationInitialize** (sound_drv_dsound_device ∗dsound_device)
- static bool soundDrvDSoundSetVolume (sound_drv_dsound_device ∗dsound_device, const int volume)
- bool **soundDrvDSoundSetCurrentSoundDeviceVolume** (const int volume)
- bool **soundDrvDSoundSetCooperativeLevel** (sound_drv_dsound_device ∗dsound_device)
- void **soundDrvDSoundPrimaryBufferRelease** (sound_drv_dsound_device ∗dsound_device)
- bool **soundDrvDSoundPrimaryBufferInitialize** (sound_drv_dsound_device ∗dsound_device)
- void **soundDrvDSoundSecondaryBufferRelease** (sound_drv_dsound_device ∗dsound_device)
- bool **soundDrvCreateSecondaryBuffer** (sound_drv_dsound_device ∗dsound_device)
- bool **soundDrvClearSecondaryBuffer** (sound_drv_dsound_device ∗dsound_device)
- bool **soundDrvInitializeSecondaryBufferNotification** (sound_drv_dsound_device ∗dsound_device)
- bool **soundDrvDSoundSecondaryBufferInitialize** (sound_drv_dsound_device ∗dsound_device)
- void **soundDrvDSoundPlaybackStop** (sound_drv_dsound_device ∗dsound_device)
- bool **soundDrvDSoundPlaybackInitialize** (sound_drv_dsound_device ∗dsound_device)
- void **soundDrvCopy16BitsStereo** (UWO ∗audio_buffer, UWO ∗left, UWO ∗right, ULO sample_count)
- void **soundDrvCopy16BitsMono** (UWO ∗audio_buffer, UWO ∗left, UWO ∗right, ULO sample_count)
- void **soundDrvCopy8BitsStereo** (UBY ∗audio_buffer, UWO ∗left, UWO ∗right, ULO sample_count)
- void **soundDrvCopy8BitsMono** (UBY ∗audio_buffer, UWO ∗left, UWO ∗right, ULO sample_count)
- bool **soundDrvDSoundCopyToBuffer** (sound_drv_dsound_device ∗dsound_device, UWO ∗left, UWO ∗right, ULO sample_count, ULO buffer_half)
- void **soundDrvPlay** (WOR ∗left, WOR ∗right, ULO sample_count)
- void **soundDrvAcquireMutex** (sound_drv_dsound_device ∗dsound_device)
- void **soundDrvReleaseMutex** (sound_drv_dsound_device ∗dsound_device)
- bool **soundDrvWaitForData** (sound_drv_dsound_device ∗dsound_device, ULO next_buffer_no, bool &need_to_restart_playback)
- bool **soundDrvProcessEndOfBuffer** (sound_drv_dsound_device ∗dsound_device, ULO current_buffer_no, ULO next_buffer_no)
- DWORD WINAPI **soundDrvThreadProc** (void ∗in)
- void **soundDrvHardReset** (void)
- bool **soundDrvEmulationStart** (ULO rate, bool bits16, bool stereo, ULO ∗sample_count_max)
- void **soundDrvEmulationStop** (void)
- bool **soundDrvStartup** (sound_device ∗devinfo)
- void **soundDrvShutdown** (void)

Variables

- sound_drv_dsound_device **sound_drv_dsound_device_current**
- volatile __int64 **timertime** = 0

11.7.1 Detailed Description

Sound driver for Windows

11.7.2 Function Documentation

11.7.2.1 soundDrvDSoundSetVolume()

```
static bool soundDrvDSoundSetVolume (
            sound_drv_dsound_device * dsound_device,
            const int volume )  [static]
```

Configure volume of secondary DirectSound buffer of current sound device.

Loudness is perceived in a logarithmic manner; the calculation attempts to utilize the upper half of the available spectrum quadratically, so that the perceived volume when moving the slider along feels more natural the numbers may still need some fine-tuning

Parameters

in	*volume*	the target volume in the range of 0 to 100 (100 being full volume)

Returns

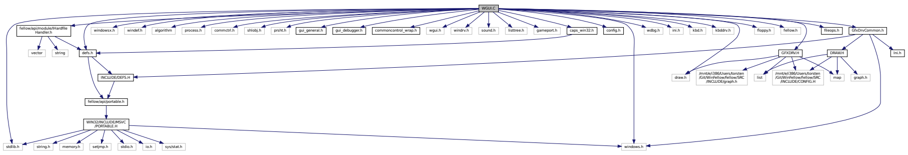
TRUE is successful, FALSE otherwise.

11.8 WGUI.C File Reference

```
#include "defs.h"
#include <windows.h>
#include <windowsx.h>
#include <windef.h>
#include <algorithm>
#include <stdlib.h>
#include <process.h>
#include <commctrl.h>
#include <shlobj.h>
#include <prsht.h>
#include "gui_general.h"
#include "gui_debugger.h"
#include "commoncontrol_wrap.h"
```

```
#include "wgui.h"
#include "windrv.h"
#include "sound.h"
#include "listtree.h"
#include "gameport.h"
#include "fellow/api/module/IHardfileHandler.h"
#include "config.h"
#include "draw.h"
#include "wdbg.h"
#include "ini.h"
#include "kbd.h"
#include "kbddrv.h"
#include "caps_win32.h"
#include "floppy.h"
#include "fellow.h"
#include "GFXDRV.H"
#include "fileops.h"
#include "GfxDrvCommon.h"
```

Include dependency graph for WGUI.C:



Macros

- #define **MAX_JOYKEY_PORT** 2
- #define **MAX_DISKDRIVES** 4
- #define **DISKDRIVE_PROPERTIES** 3
- #define **DISKDRIVE_PROPERTIES_MAIN** 4
- #define **NUMBER_OF_CHIPRAM_SIZES** 8
- #define **NUMBER_OF_FASTRAM_SIZES** 5
- #define **NUMBER_OF_BOGORAM_SIZES** 8
- #define **NUMBER_OF_SOUND_RATES** 4
- #define **NUMBER_OF_SOUND_FILTERS** 3
- #define **NUMBER_OF_CPUS** 10
- #define **NUMBER_OF_GAMEPORT_STRINGS** 6
- #define **PROP_SHEETS** 10
- #define **FILESYSTEM_COLS** 4

Typedefs

- typedef INT_PTR(CALLBACK ∗ **wguiDlgProc**) (HWND, UINT, WPARAM, LPARAM)

Enumerations

- enum { **DID_IMAGENAME**, **DID_ENABLED**, **DID_READONLY** }
- enum { **DID_IMAGENAME_MAIN**, **DID_EJECT_MAIN**, **DID_FILEDIALOG_MAIN**, **DID_LED_MAIN** }

- enum **wguiActions** {
WGUI_NO_ACTION, **WGUI_START_EMULATION**, **WGUI_QUIT_EMULATOR**, **WGUI_CONFIGURATION**,
WGUI_OPEN_CONFIGURATION, **WGUI_SAVE_CONFIGURATION**, **WGUI_SAVE_CONFIGURATION_AS**, **WGUI_LOAD_HISTORY0**,
WGUI_LOAD_HISTORY1, **WGUI_LOAD_HISTORY2**, **WGUI_LOAD_HISTORY3**, **WGUI_DEBUGGER_START**,
WGUI_ABOUT, **WGUI_LOAD_STATE**, **WGUI_SAVE_STATE** }
- enum **PROPERTYSHEETNAMES** {
PROPSHEETPRESETS = 0, **PROPSHEETCPU** = 1, **PROPSHEETFLOPPY** = 2, **PROPSHEETMEMORY** = 3,
PROPSHEETDISPLAY = 4, **PROPSHEETSOUND** = 5, **PROPSHEETFILESYSTEM** = 6, **PROPSHEETHARDFILE** = 7,
PROPSHEETGAMEPORT = 8, **PROPSHEETVARIOUS** = 9 }

Functions

- INT_PTR CALLBACK **wguiPresetDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- INT_PTR CALLBACK **wguiCPUDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- INT_PTR CALLBACK **wguiFloppyDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- INT_PTR CALLBACK wguiFloppyCreateDialogProc (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- INT_PTR CALLBACK **wguiMemoryDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- INT_PTR CALLBACK **wguiDisplayDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- INT_PTR CALLBACK **wguiSoundDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- INT_PTR CALLBACK **wguiFilesystemAddDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- INT_PTR CALLBACK **wguiFilesystemDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- INT_PTR CALLBACK **wguiHardfileCreateDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- INT_PTR CALLBACK **wguiHardfileAddDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- INT_PTR CALLBACK **wguiHardfileDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- INT_PTR CALLBACK **wguiGameportDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- INT_PTR CALLBACK **wguiVariousDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- void **wguiLoadBitmaps** (void)
- void **wguiReleaseBitmaps** (void)
- void **wguiCheckMemorySettingsForChipset** (void)
- wgui_drawmode_list & **wguiGetFullScreenMatchingList** (ULO colorbits)
- int **wguiGetDesktopBitsPerPixel** ()
- std::pair< unsigned int, unsigned int > **wguiGetDesktopSize** ()
- std::pair< unsigned int, unsigned int > **wguiGetDesktopWorkAreaSize** ()
- wgui_drawmode * **wguiGetUIDrawModeFromIndex** (unsigned int index, wgui_drawmode_list &list)
- void **wguiGetResolutionStrWithIndex** (LONG index, char char_buffer[ ])
- void **wguiGetFrameSkippingStrWithIndex** (LONG index, char char_buffer[ ])

- void **wguiSetSliderTextAccordingToPosition** (HWND windowHandle, int sliderIdentifier, int sliderText↩ Identifier, void(∗getSliderStrWithIndex)(LONG, char[ ]))
- ULO **wguiGetColorBitsFromComboboxIndex** (LONG index)
- LONG **wguiGetComboboxIndexFromColorBits** (ULO colorbits)
- DISPLAYDRIVER **wguiGetDisplayDriverFromComboboxIndex** (LONG index)
- LONG **wguiGetComboboxIndexFromDisplayDriver** (DISPLAYDRIVER displaydriver)
- void **wguiConvertDrawModeListToGuiDrawModes** ()
- void **wguiFreeGuiDrawModesLists** ()
- wgui_drawmode ∗ **wguiMatchFullScreenResolution** ()
- STR ∗ **wguiExtractFilename** (STR ∗fullpathname)
- STR ∗ **wguiExtractPath** (STR ∗fullpathname)
- static STR ∗ **wguiGetBOOLEToString** (BOOLE value)
- BOOLE **wguiSelectFile** (HWND hwndDlg, STR ∗filename, ULO filenamesize, STR ∗title, SelectFileFlags SelectFileType)
- BOOLE **wguiSaveFile** (HWND hwndDlg, STR ∗filename, ULO filenamesize, STR ∗title, SelectFileFlags SelectFileType)
- BOOLE **wguiSelectDirectory** (HWND hwndDlg, STR ∗szPath, STR ∗szDescription, ULO filenamesize, STR ∗szTitle)
- void **wguiRemoveAllHistory** (void)
- void **wguiInstallHistoryIntoMenu** (void)
- void **wguiPutCfgInHistoryOnTop** (ULO cfgtotop)
- void **wguiInsertCfgIntoHistory** (STR ∗cfgfilenametoinsert)
- void **wguiDeleteCfgFromHistory** (ULO itemtodelete)
- void **wguiSwapCfgsInHistory** (ULO itemA, ULO itemB)
- void **wguiSaveConfigurationFileAs** (cfg ∗conf, HWND hwndDlg)
- void **wguiOpenConfigurationFile** (cfg ∗conf, HWND hwndDlg)
- void **wguiSaveStateFileAs** (cfg ∗conf, HWND hwndDlg)
- void **wguiOpenStateFile** (cfg ∗conf, HWND hwndDlg)
- void **wguiInstallCPUConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiExtractCPUConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiInstallFloppyConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiInstallFloppyMain** (HWND hwndDlg, cfg ∗conf)
- void **wguiExtractFloppyConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiExtractFloppyMain** (HWND hwndDlg, cfg ∗conf)
- void **wguiInstallMemoryConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiExtractMemoryConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiInstallBlitterConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiExtractBlitterConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiInstallSoundConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiExtractSoundConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiInstallGameportConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiExtractGameportConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiInstallVariousConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiExtractVariousConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiHardfileSetInformationString** (STR ∗s, STR ∗deviceName, int partitionNumber, const Hardfile↩ Partition &partition)
- HTREEITEM **wguiHardfileTreeViewAddDisk** (HWND hwndTree, STR ∗filename, bool hasRDB, const HardfileGeometry &geometry, int hardfileIndex)
- void **wguiHardfileTreeViewAddPartition** (HWND hwndTree, HTREEITEM parent, int partitionNumber, STR ∗deviceName, const HardfilePartition &partition, int hardfileIndex)
- void **wguiHardfileTreeViewAddHardfile** (HWND hwndTree, cfg_hardfile ∗hf, int hardfileIndex)
- void **wguiInstallHardfileConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiExtractHardfileConfig** (HWND hwndDlg, cfg ∗conf)
- bool **wguiHardfileAdd** (HWND hwndDlg, cfg ∗conf, bool add, ULO index, cfg_hardfile ∗target)
- bool **wguiHardfileCreate** (HWND hwndDlg, cfg ∗conf, ULO index, cfg_hardfile ∗target)

- void **wguiFilesystemUpdate** (HWND lvHWND, cfg_filesys ∗fs, ULO i, BOOL add)
- void **wguiInstallFilesystemConfig** (HWND hwndDlg, cfg ∗conf)
- void **wguiExtractFilesystemConfig** (HWND hwndDlg, cfg ∗conf)
- BOOLE **wguiFilesystemAdd** (HWND hwndDlg, cfg ∗conf, BOOLE add, ULO index, cfg_filesys ∗target)
- void **wguiInstallDisplayScaleConfigInGUI** (HWND hwndDlg, cfg ∗conf)
- void **wguiExtractDisplayScaleConfigFromGUI** (HWND hwndDlg, cfg ∗conf)
- void **wguiInstallColorBitsConfigInGUI** (HWND hwndDlg, cfg ∗conf)
- void **wguiInstallFullScreenButtonConfigInGUI** (HWND hwndDlg, cfg ∗conf)
- void **wguiInstallDisplayScaleStrategyConfigInGUI** (HWND hwndDlg, cfg ∗conf)
- void **wguiInstallFullScreenResolutionConfigInGUI** (HWND hwndDlg, cfg ∗conf)
- void **wguiInstallDisplayDriverConfigInGUI** (HWND hwndDlg, cfg ∗conf)
- void **wguiInstallFrameSkipConfigInGUI** (HWND hwndDlg, cfg ∗conf)
- void **wguiInstallDisplayConfig** (HWND hwndDlg, cfg ∗conf)
- unsigned int **wguiDecideScaleFromDesktop** (unsigned int unscaled_width, unsigned int unscaled_height)
- void **wguiExtractDisplayFullscreenConfig** (HWND hwndDlg, cfg ∗cfg)
- void **wguiExtractDisplayConfig** (HWND hwndDlg, cfg ∗conf)
- LON **wguiListViewNext** (HWND ListHWND, ULO initialindex)
- int **wguiTreeViewSelection** (HWND hwndTree)
- void **wguiSelectDiskImage** (cfg ∗conf, HWND hwndDlg, int editIdentifier, ULO index)
- bool **wguiCreateFloppyDiskImage** (cfg ∗conf, HWND hwndDlg, ULO index)
- ULO **wguiGetNumberOfScreenAreas** (ULO colorbits)
- BOOL CALLBACK **wguiBlitterDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- void **wguiHardfileAddDialogEnableGeometry** (HWND hwndDlg, bool enable)
- void **wguiHardfileAddDialogSetGeometryEdits** (HWND hwndDlg, STR ∗filename, int sectorsPerTrack, int surfaces, int reservedBlocks, int bytesPerSector, bool enable)
- INT_PTR **wguiConfigurationDialog** ()
- void **wguiSetCheckOfUseMultipleGraphicalBuffers** (BOOLE useMultipleGraphicalBuffers)
- INT_PTR CALLBACK **wguiAboutDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- void **wguiAbout** (HWND hwndDlg)
- INT_PTR CALLBACK **wguiDialogProc** (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM l←Param)
- void **wguiRequester** (STR ∗szMessage, UINT uType)
- BOOLE **wguiCheckEmulationNecessities** (void)
- BOOLE **wguiEnter** (void)
- static bool **wguiInitializePresets** (wgui_preset ∗∗wgui_presets, ULO ∗wgui_num_presets)
- void **wguiSetProcessDPIAwareness** (const char ∗pszAwareness)
- void **wguiStartup** (void)
- void **wguiStartupPost** (void)
- void **wguiShutdown** (void)

Variables

- HWND **wgui_hDialog**
- cfg ∗ **wgui_cfg**
- ini ∗ **wgui_ini**
- STR **extractedfilename** [CFG_FILENAME_LENGTH]
- STR **extractedpathname** [CFG_FILENAME_LENGTH]
- wgui_drawmodes **wgui_dm**
- wgui_drawmode ∗ **pwgui_dm_match**
- BOOLE **wgui_emulation_state** = FALSE
- HBITMAP **power_led_on_bitmap** = 0
- HBITMAP **power_led_off_bitmap** = 0

- HBITMAP **diskdrive_led_disabled_bitmap** = 0
- HBITMAP **diskdrive_led_off_bitmap** = 0
- kbd_event **gameport_keys_events** [MAX_JOYKEY_PORT][MAX_JOYKEY_VALUE]
- int **gameport_keys_labels** [MAX_JOYKEY_PORT][MAX_JOYKEY_VALUE]
- int **diskimage_data** [MAX_DISKDRIVES][DISKDRIVE_PROPERTIES]
- int **diskimage_data_main** [MAX_DISKDRIVES][DISKDRIVE_PROPERTIES_MAIN]
- STR ∗ **wgui_chipram_strings** [NUMBER_OF_CHIPRAM_SIZES]
- STR ∗ **wgui_fastram_strings** [NUMBER_OF_FASTRAM_SIZES]
- STR ∗ **wgui_bogoram_strings** [NUMBER_OF_BOGORAM_SIZES]
- int **wgui_sound_rates_cci** [NUMBER_OF_SOUND_RATES]
- int **wgui_sound_filters_cci** [NUMBER_OF_SOUND_FILTERS]
- int **wgui_cpus_cci** [NUMBER_OF_CPUS]
- STR ∗ **wgui_gameport_strings** [NUMBER_OF_GAMEPORT_STRINGS]
- STR **wgui_preset_path** [CFG_FILENAME_LENGTH] = ""
- ULO **wgui_num_presets** = 0
- wgui_preset ∗ **wgui_presets** = NULL
- wguiActions **wgui_action**
- UINT **wgui_propsheetRID** [PROP_SHEETS]
- UINT **wgui_propsheetICON** [PROP_SHEETS]
- HWND **wgui_propsheetHWND** [PROP_SHEETS]
- wguiDlgProc **wgui_propsheetDialogProc** [PROP_SHEETS]
- static STR **FileType** [7][CFG_FILENAME_LENGTH]
- cfg_hardfile ∗ **wgui_current_hardfile_edit** = nullptr
- ULO **wgui_current_hardfile_edit_index** = 0
- cfg_filesys ∗ **wgui_current_filesystem_edit** = NULL
- ULO **wgui_current_filesystem_edit_index** = 0
- bool **wguiHardfileTreeSelecting** = false

11.8.1 Detailed Description

Window GUI code

11.8.2 Function Documentation

11.8.2.1 wguiFloppyCreateDialogProc()

```
INT_PTR CALLBACK wguiFloppyCreateDialogProc (
            HWND hwndDlg,
            UINT uMsg,
            WPARAM wParam,
            LPARAM lParam )
```

dialog procedure for creation of floppy ADF images

Returns

EndDialog is passed a pointer to the newly created floppy image, that must be freed by the caller

11.8.3 Variable Documentation

11.8.3.1 diskimage_data

```
int diskimage_data[MAX_DISKDRIVES][DISKDRIVE_PROPERTIES]
```

Initial value:

```
= {
  {IDC_EDIT_DF0_IMAGENAME, IDC_CHECK_DF0_ENABLED, IDC_CHECK_DF0_READONLY},
  {IDC_EDIT_DF1_IMAGENAME, IDC_CHECK_DF1_ENABLED, IDC_CHECK_DF1_READONLY},
  {IDC_EDIT_DF2_IMAGENAME, IDC_CHECK_DF2_ENABLED, IDC_CHECK_DF2_READONLY},
  {IDC_EDIT_DF3_IMAGENAME, IDC_CHECK_DF3_ENABLED, IDC_CHECK_DF3_READONLY}
}
```

11.8.3.2 diskimage_data_main

```
int diskimage_data_main[MAX_DISKDRIVES][DISKDRIVE_PROPERTIES_MAIN]
```

Initial value:

```
= {
  {IDC_EDIT_DF0_IMAGENAME_MAIN, IDC_BUTTON_DF0_EJECT_MAIN, IDC_BUTTON_DF0_FILEDIALOG_MAIN,
      IDC_IMAGE_DF0_LED_MAIN},
  {IDC_EDIT_DF1_IMAGENAME_MAIN, IDC_BUTTON_DF1_EJECT_MAIN, IDC_BUTTON_DF1_FILEDIALOG_MAIN,
      IDC_IMAGE_DF1_LED_MAIN},
  {IDC_EDIT_DF2_IMAGENAME_MAIN, IDC_BUTTON_DF2_EJECT_MAIN, IDC_BUTTON_DF2_FILEDIALOG_MAIN,
      IDC_IMAGE_DF2_LED_MAIN},
  {IDC_EDIT_DF3_IMAGENAME_MAIN, IDC_BUTTON_DF3_EJECT_MAIN, IDC_BUTTON_DF3_FILEDIALOG_MAIN,
      IDC_IMAGE_DF3_LED_MAIN}
}
```

11.8.3.3 FileType

```
STR FileType[7][CFG_FILENAME_LENGTH]  [static]
```

Initial value:

```
= {
  "ROM Images (.rom;.bin)\0*.rom;*.bin\0ADF Diskfiles (.adf;.adz;.adf.gz;.dms)\0*.adf;*.adz;*.adf.gz;*.dms
      \0\0\0",


  "ADF Diskfiles (.adf;.adz;.adf.gz;.dms)\0*.adf;*.adz;*.adf.gz;*.dms\0CAPS IPF Images (.ipf)\0*.ipf\0\0\0"
      ,

  "Key Files (.key)\0*.key\0\0\0",
  "Hard Files (.hdf)\0*.hdf\0\0\0",
  "Configuration Files (.wfc)\0*.wfc\0\0\0",
  "Amiga Modules (.amod)\0*.amod\0\0\0",
  "State Files (.fst)\0\0\0"
}
```

11.8.3.4 gameport_keys_events

```
kbd_event gameport_keys_events[MAX_JOYKEY_PORT][MAX_JOYKEY_VALUE]
```

Initial value:

```
= {
  {
    EVENT_JOY0_UP_ACTIVE,
      EVENT_JOY0_DOWN_ACTIVE,
      EVENT_JOY0_LEFT_ACTIVE,
      EVENT_JOY0_RIGHT_ACTIVE,
      EVENT_JOY0_FIRE0_ACTIVE,
      EVENT_JOY0_FIRE1_ACTIVE,
      EVENT_JOY0_AUTOFIRE0_ACTIVE,
      EVENT_JOY0_AUTOFIRE1_ACTIVE
  },
  {
    EVENT_JOY1_UP_ACTIVE,
      EVENT_JOY1_DOWN_ACTIVE,
      EVENT_JOY1_LEFT_ACTIVE,
      EVENT_JOY1_RIGHT_ACTIVE,
      EVENT_JOY1_FIRE0_ACTIVE,
      EVENT_JOY1_FIRE1_ACTIVE,
      EVENT_JOY1_AUTOFIRE0_ACTIVE,
      EVENT_JOY1_AUTOFIRE1_ACTIVE
    }
}
```

11.8.3.5 gameport_keys_labels

```
int gameport_keys_labels[MAX_JOYKEY_PORT][MAX_JOYKEY_VALUE]
```

Initial value:

```
= {
  { IDC_GAMEPORT0_UP, IDC_GAMEPORT0_DOWN, IDC_GAMEPORT0_LEFT, IDC_GAMEPORT0_RIGHT, IDC_GAMEPORT0_FIRE0,
      IDC_GAMEPORT0_FIRE1, IDC_GAMEPORT0_AUTOFIRE0, IDC_GAMEPORT0_AUTOFIRE1 },
  { IDC_GAMEPORT1_UP, IDC_GAMEPORT1_DOWN, IDC_GAMEPORT1_LEFT, IDC_GAMEPORT1_RIGHT, IDC_GAMEPORT1_FIRE0,
      IDC_GAMEPORT1_FIRE1, IDC_GAMEPORT1_AUTOFIRE0, IDC_GAMEPORT1_AUTOFIRE1 }
}
```

11.8.3.6 wgui_bogoram_strings

```
STR* wgui_bogoram_strings[NUMBER_OF_BOGORAM_SIZES]
```

Initial value:

```
= {
  "0 KB",
  "256 KB",
  "512 KB",
  "768 KB",
  "1024 KB",
  "1280 KB",
  "1536 KB",
  "1792 KB"
}
```

11.8.3.7 wgui_chipram_strings

```
STR* wgui_chipram_strings[NUMBER_OF_CHIPRAM_SIZES]
```

Initial value:

```
= {
  "256 KB",
  "512 KB",
  "768 KB",
  "1024 KB",
  "1280 KB",
  "1536 KB",
  "1792 KB",
  "2048 KB"
}
```

11.8.3.8 wgui_cpus_cci

```
int wgui_cpus_cci[NUMBER_OF_CPUS]
```

Initial value:

```
= {
  IDC_RADIO_68000,
  IDC_RADIO_68010,
  IDC_RADIO_68020,
  IDC_RADIO_68030,
  IDC_RADIO_68EC30,
  IDC_RADIO_68040,
  IDC_RADIO_68EC40,
  IDC_RADIO_68060,
  IDC_RADIO_68EC60,
  IDC_RADIO_68EC20
}
```

11.8.3.9 wgui_fastram_strings

```
STR* wgui_fastram_strings[NUMBER_OF_FASTRAM_SIZES]
```

Initial value:

```
= {
  "0 MB",
  "1 MB",
  "2 MB",
  "4 MB",
  "8 MB"
}
```

11.8.3.10 wgui_gameport_strings

```
STR* wgui_gameport_strings[NUMBER_OF_GAMEPORT_STRINGS]
```

Initial value:

```
= {
  "none",
  "keyboard layout 1",
  "keyboard layout 2",
  "joystick 1",
  "joystick 2",
  "mouse"
}
```

11.8.3.11 wgui_propsheetDialogProc

```
wguiDlgProc wgui_propsheetDialogProc[PROP_SHEETS]
```

Initial value:

```
= {
  wguiPresetDialogProc,
  wguiCPUDialogProc,
  wguiFloppyDialogProc,
  wguiMemoryDialogProc,
  wguiDisplayDialogProc,
  wguiSoundDialogProc,
  wguiFilesystemDialogProc,
  wguiHardfileDialogProc,
  wguiGameportDialogProc,
  wguiVariousDialogProc
}
```

11.8.3.12 wgui_propsheetHWND

```
HWND wgui_propsheetHWND[PROP_SHEETS]
```

Initial value:

```
= {
  NULL,
  NULL,
  NULL,
  NULL,
  NULL,
  NULL,
  NULL,
  NULL,
  NULL,
  NULL
}
```

11.8.3.13 wgui_propsheetICON

```
UINT wgui_propsheetICON[PROP_SHEETS]
```

Initial value:

```
= {
  0,
  IDI_ICON_CPU,
  IDI_ICON_FLOPPY,

  0,
  IDI_ICON_DISPLAY,
  IDI_ICON_SOUND,
  IDI_ICON_FILESYSTEM,
  IDI_ICON_HARDFILE,


  0,0
}
```

11.8.3.14 wgui_propsheetRID

```
UINT wgui_propsheetRID[PROP_SHEETS]
```

Initial value:

```
= {
  IDD_PRESETS,
  IDD_CPU,
  IDD_FLOPPY,
  IDD_MEMORY,
  IDD_DISPLAY,
  IDD_SOUND,
  IDD_FILESYSTEM,
  IDD_HARDFILE,
  IDD_GAMEPORT,
  IDD_VARIOUS
}
```

11.8.3.15 wgui_sound_filters_cci

```
int wgui_sound_filters_cci[NUMBER_OF_SOUND_FILTERS]
```

Initial value:

```
= {
  IDC_RADIO_SOUND_FILTER_ORIGINAL,
  IDC_RADIO_SOUND_FILTER_ALWAYS,
  IDC_RADIO_SOUND_FILTER_NEVER
}
```

11.8.3.16 wgui_sound_rates_cci

```
int wgui_sound_rates_cci[NUMBER_OF_SOUND_RATES]
```

Initial value:

```
= {
  IDC_RADIO_SOUND_15650,
  IDC_RADIO_SOUND_22050,
  IDC_RADIO_SOUND_31300,
  IDC_RADIO_SOUND_44100
}
```

Index