
86Box

86Box Project

Dec 02, 2021

USING 86BOX

1	Community	3
2	Contents	5

86Box is an IBM PC system emulator that specializes in running old operating systems and software designed for IBM PC systems and compatibles from 1981 through fairly recent system designs based on the PCI bus.

86Box is released under the GNU General Public License, version 2 or later. For more information, see the [COPYING file](#).

The project maintainer is [OBattler](#).

If you need a configuration manager for 86Box, use the [86Box Manager](#), our officially endorsed 86Box configuration manager, developed by Overdoze ([daviunic](#)).

COMMUNITY

We operate an [IRC channel](#) and a [Discord server](#) for discussing anything related to retro computing and, of course, 86Box. We look forward to hearing from you!

CONTENTS

2.1 Getting started

Here are the basic steps to help you get started with 86Box. The user interface has been designed to resemble Virtual PC, VirtualBox and other virtualizers, so if you used those programs before, this should all look familiar to you.

Step 1: Get the romset

86Box relies on a bunch of external binary files, called ROMs, to emulate various hardware. Together, these files are known as the *romset*. If you try to start 86Box without one, you'll receive an error and 86Box will close. You need to download the romset from [here](#), create a subfolder named `roms` in the folder where `86Box.exe` is located, and extract it there.

Step 2: Meet the main window

Once you got the romset in the right place, you can start `86Box.exe`. The main window has three important areas:

- **The menu bar at the top**, where most controls and options are located. See *Menu bar* for more information.
- **The display area in the middle**, which is where the display output from the emulated machine will be rendered.
- **The status bar at the bottom**, containing icons for quickly accessing the configured peripheral devices. See *Status bar* for more information.

Step 3: Configure the hardware

When you start an emulated machine, you probably want to configure it with the hardware options you want. This is much like putting together the hardware components to build a PC. To do this, go to the *Tools* menu and select *Settings*. This will bring up the *Settings* window, which has many options to choose from, split into *a handful of categories*.

Step 4: Configure the BIOS

Once you've selected the hardware components you wish to emulate, you need to make sure they're properly configured. This is done through the system BIOS, the same way it's done on a real computer. The specifics of this will of course differ from one machine to another, but generally speaking, you need to know how to enter the BIOS, which options to change, and which options to leave alone.

Step 5: Mount some images

Now that you've configured everything, you're ready to run some software in your emulated machine. Maybe you want to install an operating system or play a booter game. In any case, you'll have to mount some virtual media to get going. You can do this with the icons in the *status bar*. Icons representing removable media appear semi-transparent when their associated drive is empty, and fully opaque when media is inserted.

When you want to eject virtual media, click on the particular icon again and select *Eject* (for floppy and ZIP disks) or *Empty* (for CD-ROMs). The icon becomes semi-transparent again.

Step 6: Mouse and keyboard interaction

Now you're ready to do some stuff inside the emulated machine. Keyboard input is redirected there automatically whenever the emulator window has focus. All key presses and combinations will be redirected to the emulated machine.

Mouse input has to be manually "captured" and "released". To capture the mouse in the emulated machine, simply click inside the renderer area. Your host mouse cursor will disappear and your mouse movement and clicks will be redirected into the emulated machine. Now you can use the mouse inside the emulated machine - if the software and hardware configuration supports it, of course.

To release the mouse, press F8 and F12 simultaneously. You can also use the middle mouse button for this if the emulated mouse only has two buttons.

Step 7: What now?

If you made it this far, you got the basics of using 86Box, but there's more features and options to explore. For example, you can try out [86Box Manager](#) for easier management of multiple emulated machines. You can see what's under the *View* menu, or look at some of the more obscure options in the *Settings* window.

Keep in mind that because 86Box is constantly in development, various problems will come and go. If you think something's not working the way it should, consider submitting an issue here on GitHub or joining official support channels on Discord or IRC.

Have fun!

2.2 Menu bar

The menu bar located at the top of the 86Box window provides controls for the emulated machine as a whole, its display, and the 86Box user interface.

2.2.1 Action

- **Keyboard requires capture:** require the mouse to be captured for keypresses to be forwarded to the emulated machine. Enabling this option allows the use of keyboard combinations (such as Alt+Tab) on the host system while 86Box is focused.
- **Right CTRL is left ALT:** let the right Ctrl key act as a left Alt key, to simulate some special keyboards where the Alt key is located on the right side of the space bar.
- **Hard Reset:** force a reset of the emulated machine. Requires confirmation, which can be disabled by checking the *Don't show this message again* box.
- **Ctrl+Alt+Del:** send a *Ctrl+Alt+Del* key combination to the emulated machine. You can alternatively press *Ctrl+F12* to send that combination.

- **Ctrl+Alt+Esc:** send a *Ctrl+Alt+Esc* key combination to the emulated machine.
- **Pause:** pause emulation. Uncheck this option to resume emulation.

Note: Emulation is automatically paused when the emulated machine enters ACPI sleep mode.

- **Exit:** quit 86Box. Requires confirmation, which can be disabled by checking the *Don't show this message again* box.

2.2.2 View

- **Resizable window:** allow the 86Box window to be freely resized. Unchecking this option will also return the window to its normal size.
- **Remember size & position:** automatically save the size and position of the 86Box window to the emulated machine's configuration file.
- **Renderer:** select a graphical renderer for the emulated display. *SDL (Hardware)* is recommended in most cases. *OpenGL (3.3 Core)* allows for shader effects to be applied to the emulated display, however, it is not compatible with older integrated GPUs.
- **OpenGL options:** configure the *OpenGL (3.3 Core)* renderer. This submenu will be available if that renderer is selected.
 - **Target framerate:** select the framerate at which the emulated display is updated. *Sync with video* uses the emulated display's current refresh rate.
 - **VSync:** enable vertical sync. Recommended if tearing artifacts are observed.
 - **Select shader:** load a *.glsl* shader file to apply on the emulated display. Many shaders are available for simulating CRT displays, VHS tapes and other aesthetics; the [RetroArch glsl-shaders repository](#) is a good place to start.
 - **Remove shader:** disable the currently-loaded shader.

Note: Shaders that take advantage of multipass and previous frames are not supported.

- **Specify dimensions:** open a window where an exact size (in pixels) for the emulated display can be set. If checked, the *Lock to this size* box prevents changes in the emulated display's resolution from overriding the specified size.
- **Force 4:3 display ratio:** stretch the emulated display to a 4:3 aspect ratio, independently of the emulated machine's screen resolution.
- **Window scale factor:** scale the emulated display to half (0.5x), normal (1x), 50% larger (1.5x) or double (2x) sizes.
- **Filter method:** select the filtering method (Nearest or Linear) to be used when scaling the emulated display.
- **HiDPI scaling:** automatically scale the emulated display to real size if your host system has a HiDPI display. This option can be used alongside *Window scale factor* above.
- **Fullscreen:** enter full screen mode. Press *Ctrl+Alt+Page Down* to go back to windowed mode. You can also enter full screen mode by pressing *Ctrl+Alt+Page Up*.
- **Fullscreen stretch mode:** select the picture mode to use when in full screen mode.
 - **Full screen stretch:** stretch the emulated display to completely fill the host display.

- **4:3:** stretch the emulated display to a 4:3 aspect ratio, then scale it to fit the host display.
- **Square pixels (keep ratio):** scale the emulated display to fit the host display, without changing the aspect ratio.
- **Integer scale:** scale the emulated display to the largest integer scale factor to fit the host display. This provides the highest possible picture quality, at the cost of black bars if the host display's resolution is not divisible by the emulated display's resolution.
- **EGA/(S)VGA settings:** contains display settings specific to EGA, VGA and Super VGA video hardware.
 - **Inverted VGA monitor:** emulate a VGA monitor with inverted colors.
 - **VGA screen type:** select the VGA monitor type to emulate. Color, grayscale, amber phosphor, green phosphor and white phosphor monitors can be selected.
 - **Grayscale conversion type:** select the color-to-grayscale conversion profile (BT.601, BT.709 or Average) to use when a grayscale monitor is selected.
- **CGA/PCjr/Tandy/EGA/(S)VGA overscan:** add an overscan border around the display. This border is only added when emulating the specified video hardware types.
- **Change contrast for monochrome display:** optimize the contrast of monochrome CGA monitors for 4-color operation.

2.2.3 Media

This menu lists all storage drives attached to the emulated machine, and provides the same controls that are accessible by clicking the respective drive's icon on the *status bar*.

2.2.4 Tools

- **Settings:** open the *Settings* window.
- **Update status bar icons:** enable the activity lights on *status bar* icons. Unchecking this option may improve emulation performance on low-end host systems.
- **Preferences:** open the *Preferences* window, which provides the following options:
 - **Language:** select a language for the 86Box user interface.
 - **Icon set:** select an icon theme for the *status bar* and *Settings window*.
- **Enable Discord integration:** enable Discord Rich Presence. 86Box shares the emulated machine's name, model and CPU with other Discord users.

Note: Discord integration will not be available if the Discord desktop app is not running, or if the included `discord_game_sdk.dll` file is missing from the 86Box directory.

- **Take screenshot:** take a screenshot of the emulated display. Screenshots are saved as .png images in the `screenshots` subdirectory found in the emulated machine's directory.
- **Sound gain:** open the *sound gain control*, which is also accessible through the status bar.

2.2.5 Help

- **Documentation:** open the very documentation you're reading.
- **About 86Box:** show credits and license information about 86Box.

2.3 Status bar

The status bar located at the bottom of the 86Box window provides icons related to devices attached to the emulated machine. Move your mouse cursor over an icon to see what device it represents. **Most icons can be clicked on** to access options related to their respective devices, which are listed below. Additionally, a green indicator light will appear on an icon when its device is in use, unless *Update status bar icons* is disabled.

2.3.1 Cassette deck

A cassette tape icon will appear if *IBM cassette emulation* is enabled.

- **New image:** create a new cassette tape image file.
- **Existing image:** insert a *cassette tape image file* into the deck.
- **Existing image (Write-protected):** insert a cassette tape image file into the deck as a read-only tape.
- **Record:** start recording data to the cassette tape. Not available if the tape is read-only.
- **Play:** start playing the cassette tape.
- **Rewind to the beginning:** rewind the cassette tape to its beginning.
- **Fast forward to the end:** fast forward the cassette tape to its end.
- **Eject:** remove the currently-inserted cassette tape from the deck.

2.3.2 PCjr cartridges

Two cartridge icons will appear if the **IBM PCjr** is being emulated. Each icon corresponds to a cartridge slot on the PCjr's front panel.

- **Image:** insert a *cartridge image file* into this slot. Inserting a cartridge will reset the PCjr.
- **Eject:** remove the currently-inserted cartridge from this slot.

2.3.3 Floppy drives

A 3.5" or 5.25" floppy icon will appear for each configured *floppy drive*.

- **New image:** create a new disk image file. Opens the *New Image* window, which lets you select the image size and where to save the file.
- **Existing image:** insert a *disk image file* into this drive.
- **Existing image (Write-protected):** insert a disk image file into this drive as a read-only disk.

- **Export to 86F:** convert the currently-inserted disk image file to 86Box's *86F* surface image format. You will be asked where to save the converted file.
- **Eject:** remove the currently-inserted disk from this drive.

2.3.4 CD-ROM drives

A CD icon will appear for each configured *CD-ROM drive*.

- **Mute:** mute any *CD audio* played through this drive's analog output. CD audio is unmuted by default on the first configured CD-ROM drive.
- **Empty:** remove any disc inserted into this drive.
- **Reload previous image:** reinsert the last disc image file selected through the *Image* option.
- **Image:** insert a *CD-ROM or DVD-ROM disc image file* into this drive.

2.3.5 ZIP and MO drives

A ZIP or MO icon will appear for each configured *additional removable storage drive*.

- **New image:** create a new disk image file. Opens the *New Image* window, which lets you select the image size and where to save the file.
- **Existing image:** insert a *disk image file* into this drive.
- **Existing image (Write-protected):** insert a disk image file into this drive as a read-only disk.
- **Eject:** remove the currently-inserted disk from this drive.
- **Reload previous image:** reinsert the last disk image file selected through the *Existing image* options.

2.3.6 Hard disks

A hard disk icon will appear for each configured *hard disk bus*. For example, if you have both IDE and SCSI hard disks configured, two hard disk icons will appear: one representing all IDE disks, and another one representing all SCSI disks. No options are available.

2.3.7 Network

This icon will appear if *networking* is enabled. No options are available.

2.3.8 Sound

This icon is always present. Double-clicking it opens a sound gain control, which allows you to increase the loudness of all audio produced by the emulated machine's PC speaker, *sound cards* and other sound hardware.

Note: The gain control does not apply to MIDI music sent to a software synthesizer through the *System MIDI* device, as these synthesizers are external to 86Box.

2.3.9 Additional information area

This area, located to the right of the icons described above, contains additional information which may be provided by components such as the *ISABugger* and *POST card*.

Monitor sleep mode

The *Monitor in sleep mode* message will be displayed if the emulated monitor has been put into DPMS sleep mode by the operating system. Pressing a key or moving the mouse is often enough to wake the monitor up.

ISABugger

The ISABugger's hexadecimal displays and LED banks are displayed here. See *ISABugger* for more information.

POST card

The leftmost hexadecimal value is the most recent POST code reported, while the rightmost value is the second most recent code, like on a real dual-display POST card. A value of -- indicates that no POST code has been reported yet.

Note: The additional information area can only be used by one component at a time. If both the ISABugger and the POST card are enabled simultaneously, the POST card takes over whenever a POST code is reported, and the ISABugger takes over whenever one of its registers is written to. The *Monitor in sleep mode* message is high-priority and will override all other components.

2.4 Settings

The *Settings* window allows you to configure the emulated machine.

If any changes were made to the settings, you will be asked whether or not you want to save the changes upon pressing *OK* or closing the window. Saving changes will hard reset the emulated machine. Press *Cancel* to immediately discard all changes.

2.4.1 Machine

The *Machine* page contains settings related to the emulated machine as a whole, such as the machine type, CPU type and amount of memory.

Machine type / Machine

Machine/motherboard model to emulate. The *Machine* box lists all available models for the machine class selected on the *Machine type* box.

The *Configure* button opens a new window with settings specific to the machine's onboard devices, such as the amount of installed video memory for an onboard video chip.

CPU type / Speed

Main processor to emulate. The *Speed* box lists all available speed grades for the processor family selected on the *CPU type* box. These boxes only list processor types and speed grades supported by the machine selected above.

FPU

Math co-processor to emulate. This box is not available if the processor selected above has an integrated co-processor or lacks support for an external one.

Wait states

Number of memory wait states to use on a 286- or 386-class processor. This box is not available if any other processor family is selected above.

Memory

Amount of RAM to give the emulated machine. The minimum and maximum allowed amounts of RAM will vary depending on the machine selected above.

Dynamic Recompiler

Enable the dynamic recompiler, which provides faster but less accurate CPU emulation. The recompiler is available as an option for 486-class processors, and is mandatory starting with the Pentium.

Time synchronization

Automatically copy your host system's date and time over to the emulated machine's hardware real-time clock. Synchronization is performed every time the emulated operating system reads the hardware clock to calibrate its own internal clock, which usually happens once on every boot.

- **Disabled:** do not perform time synchronization. The emulated machine's date and time can be set through its operating system or BIOS setup utility. Time only ticks while the emulated machine is running.
- **Enabled (local time):** synchronize the time in your host system's configured timezone. Use this option when emulating an operating system which stores *local time* in the hardware clock, such as DOS or Windows.

- **Enabled (UTC):** synchronize the time in Coordinated Universal Time (UTC). Use this option when emulating an operating system which stores *UTC time* in the hardware clock, such as Linux.

2.4.2 Display

The *Display* page contains settings related to the emulated machine's 2D and 3D video cards.

Video

Video card to emulate. This box only lists cards supported by the machine's expansion buses. On machines equipped with an onboard video chip, the *Internal* option enables the onboard video.

The *Configure* button opens a new window with settings specific to the selected video card, such as the amount of video memory.

Voodoo Graphics

Emulate a **3dfx Voodoo** add-on 3D accelerator, connected to both the PCI bus and the video card selected above.

The *Configure* button provides the following settings:

- **Voodoo type:** type of Voodoo card to emulate.
 - **Voodoo Graphics:** the original Voodoo model, with a single Texture Mapping Unit operating at 50 MHz.
 - **Obsidian SB50 + Amethyst:** a variant of the Voodoo Graphics, with two Texture Mapping Units operating at 50 MHz.
 - **Voodoo 2:** the second Voodoo model, with two Texture Mapping Units operating at 90 MHz, as well as SLI support.

Note: The **Voodoo Banshee** and **Voodoo 3** are independent video cards, which are not found here; they must be selected on the *Video* box above. For these cards, the *Configure* button next to the *Video* box provides similar settings to the ones listed here.

- **Framebuffer memory size / Texture memory size:** amount of video memory for the Frame Buffer Interface and Texture Mapping Unit(s), respectively.
- **Bilinear filtering:** apply bilinear filtering to smooth out textures displayed on screen.
- **Screen Filter:** apply a filter to make the screen picture resemble the DAC (digital-to-analog converter) output of a real Voodoo card.
- **Render threads:** split the workloads of each Voodoo card into different CPU threads for faster emulation. The recommended amount of render threads depends on your host system's CPU core count, and whether or not Voodoo 2 SLI is enabled:

Host cores	Recommended render threads	
	Single card	Voodoo 2 SLI
2	1	1
4	2	1
6 or 8	4	2
10 or more	4	4

- **SLI:** add a second Voodoo 2 card to the system, connected to the first one through a Scan Line Interleave (SLI) interface.
- **Recompiler:** enable the Voodoo recompiler for faster emulation.

2.4.3 Input devices

The *Input devices* page contains settings related to the emulated machine's mouse, joysticks and other input devices.

Mouse

Emulate a pointing device. The following mouse types are supported:

- **Bus mouse:** ISA expansion card with a mouse interface. The I/O port and IRQ used by the card are configurable.
- **Serial mouse:** connected to the serial port of your choosing. The selected serial port must be enabled on the *Ports tab*.
- **PS/2 mouse:** connected to the PS/2 port. Only available on machines with a PS/2 mouse port.

The *Configure* button opens a new window with settings specific to the selected mouse type, such as the number of buttons, or the serial port for a serial mouse.

Joystick

Emulate one or more game port controller(s). The following controller types are supported:

- **None:** no controller connected.
- **Standard 2-button joystick(s):** up to two controllers, each with two buttons and an analog stick.
- **Standard 4-button joystick:** single controller with four buttons and an analog stick.
- **Standard 6-button joystick:** single controller with four regular buttons, two additional buttons mapped to the third and fourth axes, and an analog stick.
- **Standard 8-button joystick:** single controller with four regular buttons, four additional buttons mapped to the third and fourth axes, and an analog stick.
- **4-axis 4-button joystick:** single controller with four buttons and two analog sticks (or four axes).
- **CH Flightstick Pro:** flight controller with four buttons, three axes and a POV hat.
- **Microsoft SideWinder Pad:** up to four controllers, each with 10 buttons and a directional pad. Not compatible with standard game port joysticks; requires a driver in the emulated machine.
- **Thrustmaster Flight Control System:** flight controller with four buttons, two axes and a POV hat.

Note: A generic game port card is emulated if the machine has no game ports (either built-in or as part of a Plug and Play or PCI sound card) to accommodate the selected controller. This generic card uses the default I/O ports 200-207h, which can be changed through ISA Plug and Play. On IBM PS/1 machines, the generic card uses port 201h only and has no Plug and Play capability.

Joystick 1-4...

Configure the mappings for each emulated game port controller. The *Device* box lists all game controllers connected to the host, while the other boxes allow you to map each axis or button of the emulated controller to the real controller.

If you're not sure as to what axis or button numbers map to which sticks and buttons on the real controller, use the *Test* feature of Windows' *Game Controllers* control panel (`joy.cpl`). Keep in mind 86Box's button numbers start with 0, whereas the control panel's numbers start with 1.

Note: XInput controllers are accessed through their DInput emulation mode at the moment.

2.4.4 Sound

The *Sound* page contains settings related to the emulated machine's audio hardware.

Parallel port sound devices such as the **Disney Sound Source** and **Covox Speech Thing** are not present on this page; they can be configured through the *Ports page*.

Sound card

Sound card to emulate. Only cards supported by the machine's expansion buses will be listed. On machines equipped with an onboard sound chip, the *Internal* option enables the onboard sound.

The *Configure* button opens a new window with settings specific to the selected sound card, such as the I/O ports, IRQ and DMA channels for ISA cards.

Emulation for the Yamaha OPL series of synthesizers (used by many of the emulated cards) is provided by a modified [Nuked OPL3](#) library.

MIDI Out Device

Device to output MIDI music to, for sound cards equipped with an external MIDI output.

- **None:** don't output MIDI music.
- **FluidSynth:** a software soundfont synthesizer. Selecting a soundfont file is required. There will be no synthesizer output if no soundfont is configured or if the included `libfluidsynth.dll` or `libfluidsynth64.dll` file is missing from the 86Box directory.
- **Roland MT-32/CM-32L Emulation:** emulate a Roland synthesizer module. Emulation is provided by the [Munt](#) library.
- **System MIDI:** output to a MIDI device on the host system, such as the Windows software synthesizer or a USB MIDI adapter.

The *Configure* button opens a new window with settings specific to the selected output device, such as the soundfont to use for *FluidSynth* and the host MIDI device to use for *System MIDI*.

MIDI In Device

Device to receive MIDI music from, for sound cards equipped with an external MIDI input.

- **None:** don't receive MIDI music.
- **System MIDI:** receive from a MIDI device on the host system, such as a USB MIDI adapter.

The *Configure* button opens a new window with settings specific to the selected input device, such as the host MIDI device to use for *System MIDI*.

Standalone MPU-401

Emulate a standalone **Roland MIDI Processing Unit** ISA card, which allows for MIDI input and output without a MPU-401-equipped sound card.

The I/O port and IRQ can be configured through the *Configure* button.

Innovation SSI-2001

Emulate the **Innovation SSI-2001** ISA sound card, based on the MOS Technology 6581 chip (commonly known as the Commodore SID) and supported by a limited number of games.

SID emulation is provided by the [reSID](#) library.

CMS / Game Blaster

Emulate the **Creative Music System** or **Game Blaster** ISA sound card, based on dual Philips SAA1099 chips and supported by some games.

Gravis Ultrasound

Emulate the **Gravis UltraSound** ISA sound card.

The type of UltraSound to emulate (Classic or MAX), I/O port and amount of onboard RAM can be configured through the *Configure* button.

Use FLOAT32 sound

Use the 32-bit floating point (instead of 16-bit integer) data type for audio output, which is less prone to clipping but may not work at all on some host systems. Try disabling this if you're getting no audio output from 86Box at all.

2.4.5 Network

The *Network* page contains settings related to the emulated machine's network connectivity.

Network type

Network emulation mode to use. See *Networking* for more information on these.

- **None:** disable networking.
- **PCap:** connects directly to a host network adapter. Similar to the **Bridge** mode on other emulators and virtualizers.
- **SLiRP:** creates a private network with a virtual router. Similar to the **NAT** mode on other emulators and virtualizers.

PCap device

Host network adapter to use for PCap mode. If no adapters appear on this list, make sure that:

- A WinPcap-compatible driver is installed;
- The installed driver is compatible with your version of Windows;
- At least one compatible (wired) network adapter is present.

Network adapter

Network card to emulate. Only cards supported by the machine's expansion buses will be listed.

The *Configure* button opens a new window with settings specific to the selected network card, such as the I/O port and IRQ for ISA cards.

The **[LPT] Parallel Port Internet Protocol** network adapter requires a **PLIP Network** device to be attached to a *parallel port*.

2.4.6 Ports (COM & LPT)

The *Ports (COM & LPT)* page contains settings related to the emulated machine's I/O ports.

LPT1-3 Device

Emulated device to connect to the given parallel (LPT) port.

- **None:** no device connected.
- **Disney Sound Source:** sound device with a resistor ladder DAC (digital-to-analog converter) and FIFO, supported by many games.
- **LPT DAC / Covox Speech Thing:** sound device with a simple resistor ladder DAC, supported by many games (through compatibility with the *Disney Sound Source* above), demos and trackers.
- **Stereo LPT DAC:** stereo version of the LPT DAC, using the *Strobe* pin to select the active output channel.
- **Generic Text Printer:** simple printer capable of outputting text only.
 - Printed documents are saved as .txt files in the `printer` subdirectory found in the emulated machine's directory.
- **Generic ESC/P Dot-Matrix:** EPSON ESC/P-compatible printer.
 - Printed pages are saved as .png files in the `printer` subdirectory found in the emulated machine's directory.

- Use the **EPSON LQ-2500** printer driver for best results.
- This printer will not work if the included `freetype.dll` file is missing from the 86Box directory.
- **Generic PostScript Printer:** PostScript-compatible printer with PDF output.
 - Printed documents are saved as `.ps` files in the `printer` subdirectory found in the emulated machine's directory. These files are automatically converted to `.pdf` once printing is completed.
 - The original `.ps` files may remain in the directory if PDF conversion fails, or if the included `gsd1132.dll` or `gsd1164.dll` file is missing from the 86Box directory.
 - Use the generic PostScript printer driver provided by your operating system.
 - Windows 95 and newer do not have a generic PostScript driver; use the **Apple LaserWriter IIf** driver for grayscale, or the **Apple Color LW 12/660 PS** driver for color.
- **PLIP Network:** A **Parallel Line Internet Protocol** cable connected to the *emulated network*.
 - The *emulated network adapter* must also be set to **[LPT] PLIP**.
 - PLIP is compatible with the DOS `plip.com` packet driver and the Linux `plip` driver (only with interrupts enabled). It is not compatible with the Windows *Direct Cable Connection* feature or any other parallel port networking implementations.
 - PLIP works best with the **SLiRP network type** due to its point-to-point nature.

Serial port 1-4

Enable emulation of serial ports ranging from COM1 to COM4. Any ports not provided by the machine's motherboard will be emulated as generic ISA or VLB serial cards.

Parallel port 1-3

Enable emulation of parallel ports ranging from LPT1 to LPT3. Any ports not provided by the machine's motherboard will be emulated as generic ISA or VLB parallel cards.

2.4.7 Storage controllers

The *Storage controllers* page contains settings related to the emulated machine's disk drive controllers.

HD Controller

Hard disk drive controller card to emulate. This box only lists cards supported by the machine's expansion buses. MFM, RLL, ESDI and IDE controllers are available. Selecting an IDE controller is not required for machines with onboard IDE.

The *Configure* button opens a new window with settings specific to the selected controller card, such as the BIOS option ROM address.

FD Controller

Floppy disk drive controller card to emulate. Selecting a controller is not required, unless you wish to use one of the add-on controllers for adding high-density 1.44M floppy support to XT machines.

The BIOS option ROM address used by the selected controller can be configured through the *Settings* button.

Tertiary / Quaternary IDE Controller

Add a third or fourth (respectively) IDE channel to the emulated machine, through a generic ISA or VLB IDE controller card.

The IRQ used by each controller can be configured through its respective *Settings* button.

Note: The tertiary and quaternary controllers are not Plug and Play compliant by default; they may require manual configuration of emulated operating systems, and may not be bootable. See *Tertiary and quaternary IDE* for more information.

SCSI Controllers

SCSI host bus adapter cards to emulate. Up to 4 SCSI cards are supported. The selection boxes only list cards supported by the machine's expansion buses.

The *Configure* buttons open a new window with settings specific to the corresponding SCSI card, such as the I/O port and IRQ for ISA cards.

Cassette

Enable IBM cassette tape emulation. The cassette deck can be controlled through the *status bar* or *Media menu*.

2.4.8 Hard disks

The *Hard disks* page contains settings related to the emulated machine's fixed disks.

Hard disk list

All hard disks attached to the emulated system are listed, with the following information:

- **Bus:** storage bus the disk is attached to, as well as the disk's bus channel or ID. These can be changed through the *Bus* and *Channel/ID* boxes below the list.
- **File:** path to the disk image file.
- **C/H/S:** disk size in cylinders, heads and sectors, respectively.
- **MB:** disk size in megabytes.

Adding a new disk

The *New...* button opens a new window allowing you to create an existing hard disk image file.

- **File name:** where to save the disk image file. See *Hard disk images* for a list of supported image formats.
- **Cylinders/Heads/Sectors:** CHS parameters for the disk image. These boxes control the *Size (MB)* box below.
- **Size (MB):** the disk image's size in MB. This box controls the *Cylinders*, *Heads* and *Sectors* boxes above. There are limits to how big a hard disk image can be; see *Hard disk size limits* for more information.
- **Bus:** storage bus to attach the disk to.
- **Channel/ID:** where to attach the disk on the selected storage bus.
 - On IDE disks, the first number corresponds to the IDE channel, and the second number corresponds to the Master/Slave position:

Value	Channel	Device
0:0	Primary	Master
0:1	Primary	Slave
1:0	Secondary	Master
1:1	Secondary	Slave
2:0	Tertiary	Master
2:1	Tertiary	Slave
3:0	Quaternary	Master
3:1	Quaternary	Slave

- On SCSI disks, the first number corresponds to the *SCSI controller* (starting at 0 instead of 1), and the second number is the SCSI ID within that controller:

Value	Controller	SCSI ID
0:00	Controller 1	0
0:15		15
1:00	Controller 2	0
1:15		15
2:00	Controller 3	0
2:15		15
3:00	Controller 4	0
3:15		15

- On MFM/RLL, XTA and ESDI disks, the second number is 0 for the first drive on the controller, and 1 for the second drive.

Note: If the disk is attached to a channel or controller that doesn't exist, such as the tertiary IDE channel with no tertiary IDE controller present, it will be effectively disabled.

Press the *OK* button to create the disk image file, or *Cancel* to close the window.

Adding an existing disk

The *Existing...* button opens a similar window to the *New...* button, except that it lets you select an existing disk image file. The CHS parameters are guessed from the image's file size, or the file header if the image is of a format which contains a header.

After selecting the image file and checking if the parameters are correct, select the *Bus* and *Channel/ID* for the hard disk and press *OK* to add it. Press *Cancel* to close the window.

Removing a disk

Select a disk on the list and press *Remove* to remove it.

2.4.9 Floppy & CD-ROM drives

The *Floppy & CD-ROM drives* page contains settings related to the emulated machine's base removable storage drives.

Floppy drives

Up to four floppy disk drives can be attached to the emulated system, although not all machines provide BIOS support for more than two drives. The following settings apply to the selected drive:

- **Type:** floppy drive to emulate. Some types have special properties and should only be used in very specific applications:
 - **5.25" 1.2M PS/2** and **3.5" 1.44M PS/2:** IBM PS/2 drives, which invert the polarity of the Density Select pin.
 - **5.25" 1.2M 300/360 RPM** and **3.5" 1.44M 300/360 RPM:** "3-mode" drives, which are capable of reading 360K 5.25" or NEC PC-98 3.5" disks if the emulated machine's BIOS supports 3-mode operation.
 - **3.5" 1.44M PC-98:** NEC PC-98 drive, which is 3-mode and inverts the polarity of the Density Select pin.
- **Turbo timings:** run the drive mechanism as fast as possible. This decreases access times and makes some incorrectly-dumped floppies readable, but may cause issues with some operating systems and applications.
- **Check BPB:** if unchecked, 86Box will ignore the [DOS BIOS Parameter Block](#) when determining the physical media format for a floppy image on this drive. See [Floppy disk detection](#) for more details.

Note: Disabling "Check BPB" may be required in order to access UNIX/Linux installation floppies or other non-DOS disks, as outlined on [Floppy disk detection](#).

Floppy disk images can be inserted and removed through the *status bar* or *Media menu*.

CD-ROM drives

Up to four CD-ROM / DVD-ROM optical disc drives can be attached to the emulated system. The following settings apply to the selected drive:

- **Bus:** storage bus to attach the drive to. ATAPI (IDE) and SCSI are supported.
- **Channel/ID:** where to attach the drive on the selected storage bus. See [Adding a new disk](#) for more information.
- **Speed:** maximum transfer speed for the drive. Up to 72x is supported.

CD-ROM / DVD-ROM disc images can be inserted and removed through the *status bar* or *Media menu*.

2.4.10 Other removable devices

The *Other removable devices* page contains settings related to the emulated machine's additional removable storage drives.

MO / ZIP drives

Up to four Magneto-Optical and four Iomega ZIP disk drives can be attached to the emulated system. The following settings apply to the selected drive:

- **Bus:** storage bus to attach the drive to. ATAPI (IDE) and SCSI are supported.
- **Channel/ID:** where to attach the drive on the selected storage bus. See [Adding a new disk](#) for more information.
- **Type (MO only):** drive model to identify as. A list of drive models to choose from is provided.
- **ZIP 250 (ZIP only):** enable the drive to read and write 250 MB ZIP disks.

MO / ZIP disk images can be inserted and removed through the *status bar* or *Media menu*.

2.4.11 Other peripherals

The *Other peripherals* page contains settings related to disk drive controllers, memory expansions and other expansion cards.

ISA RTC

Emulate an ISA real-time clock card, for machines without an integrated real-time clock.

The I/O port and/or IRQ used by the selected controller can be configured through the *Settings* button.

ISA Memory Expansion

Add up to four ISA-based memory expansion cards, for machines which support memory expansion through the ISA bus.

The memory start address and size for each expansion card can be configured through its respective *Settings* button.

ISABugger

Emulate an **ISABugger** debugging interface card, equipped with two hexadecimal displays and two LED banks, all controlled by the emulated machine. See *ISABugger* for documentation on the card's features.

POST card

Emulate a diagnostic POST card, which displays POST code values issued by the emulated machine's BIOS on the status bar. See *Status bar: POST card* for more information.

The POST card will automatically use the correct diagnostic I/O port for the emulated machine:

- Port 10h on the IBM PCjr;
- Port 60h on the IBM XT;
- Port 80h on the IBM AT, clones and the XT-based Xi 8088;
- Port 84h on early Compaq machines;
- Port 0190h on IBM PS/1 and PS/2 machines not based on the Micro Channel Architecture;
- Port 0680h on Micro Channel Architecture machines.

Note: Some operating systems and applications use port 80h (which is shared with the POST card on most machines) for other purposes. If you notice the POST code display is flickering and the emulation speed has decreased drastically, try disabling the POST card.

2.5 Machine-specific notes

This page contains important notes related to specific machine models emulated by 86Box.

2.5.1 80286

IBM AT

- The IBM Personal Computer Diagnostics disks are not Y2K-compliant and will produce a *0152 ERROR - SYSTEM BOARD* code if *time synchronization* is enabled. This code can be cleared by disabling time synchronization, then clearing the CMOS by deleting `ibmat.nvr` from the `nvr` directory.

2.5.2 Socket 7

ASUS P/I-P65UP5 (C-P55T2D)

- Modular motherboard, consisting of a **P/I-P65UP5** baseboard and one of the following CPU cards:
 - **C-P55T2D**: Socket 7 with Intel 430HX northbridge.
 - **C-P6ND**: Socket 8 with Intel 440FX northbridge.
 - **C-PKND**: Slot 1 with Intel 440FX northbridge.
- While the northbridge depends on the selected CPU card, the southbridge always remains the Intel PIIX3, as it is located on the baseboard.
- The real CPU cards support dual CPUs. As 86Box does not emulate multiprocessing, only a single CPU will be present.
- Due to a lack of I/O APIC emulation at the moment, 86Box will patch the MultiProcessor Specification tables out of RAM during boot, so that operating systems will not hang or exhibit other erratic behavior due to the missing I/O APIC.

2.5.3 Socket 8

ASUS P/I-P65UP5 (C-P6ND)

See: *ASUS P/I-P65UP5 (C-P55T2D)*

2.5.4 Slot 1

ASUS P/I-P65UP5 (C-PKND)

See: *ASUS P/I-P65UP5 (C-P55T2D)*

A-Trend ATC6310BXII

- Equipped with the obscure SMSC Victory66 southbridge instead of the regular Intel PIIX4E.
 - The Victory66 has faster IDE - up to Ultra ATA/66 as opposed to the PIIX4E's Ultra ATA/33 - and a different USB controller.
 - Drivers for Windows 95, 98, Me and 2000 are available [here](#). Windows XP, Vista and 7 include drivers out of the box.

2.5.5 Slot 2

Gigabyte GA-6GXU

- The BIOS display will corrupt itself during the memory test if the maximum of 2048 MB RAM is selected. This is a visual glitch which does not otherwise negatively impact the machine.

Freeway FW-6400GX

- Hybrid motherboard supporting both Slot 1 and Slot 2 CPUs.
- The maximum amount of RAM is limited to 2032 MB due to a BIOS bug with 2048 MB.
- ACPI is disabled by default. It can be enabled through the *ACPI Aware O/S* option of the *Power Management Setup* menu on the BIOS setup.
- Once enabled, ACPI *does not work correctly* if a non-Intel CPU is selected.

2.5.6 Socket 370

A-Trend ATC7020BXII

See: *A-Trend ATC6310BXII*

AEWIN AW-O671R

- Equipped with dual Winbond W83977EF Super I/O chips driving four serial (COM1-COM4) and two parallel (LPT1-LPT2) ports.
 - The I/O ports and IRQs used by all these ports can be configured in the BIOS setup.
- ACPI is disabled by default, unlike other machines with Award v6.00PG BIOS. It can be enabled through the *ACPI function* option of the *Power Management Setup* menu on the BIOS setup.

2.5.7 Miscellaneous

Microsoft Virtual PC 2007

- This machine loads the American Megatrends BIOS from Virtual PC 2007 into 86Box's emulation. It does not use the virtualization engine or any other components from Virtual PC.
- Virtual PC's special 8 MB video card, network card, WDM sound card and Guest Additions are not emulated by 86Box.

2.5.8 Footnotes

Broken ACPI

Some machines may have faulty or otherwise incomplete [Advanced Configuration and Power Interface](#) implementations in their BIOSes, symptoms of which include:

- Windows 2000 and higher will install the "Standard PC" HAL, which does not enable ACPI features such as soft power off and sleep mode;
- Booting an existing Windows installation with the ACPI HAL will result in a STOP 0x000000A5 blue screen;
- Booting Windows Vista or 7 (which require ACPI) will also result in a STOP 0x000000A5 blue screen, or a Windows Boot Manager 0xc0000225 error.

There is no solution to this issue, as none of the currently emulated machines with broken ACPI ever received a BIOS update to fix it.

2.6 Disk images

86Box supports a large variety of disk image formats for the emulated storage drives.

2.6.1 Hard disk images

Supported formats:

Format	File extension
Raw image	Many *
Japanese FDI	.hdi
<i>Extended HDI (HDX)</i>	.hdx
Virtual Hard Disk	.vhd

* Raw image extensions recognized by 86Box include: .hdd .ima .img

Hard disk size limits

There are limits to how big of a hard disk an emulated machine can accept. Such limits will vary depending on the machine's BIOS. The table below lists all important limits applicable to the IDE bus:

Limit	Disk size	Cylinders	Heads	Sectors
20-bit CHS	504 MB	1024	16	63
12-bit cylinder	2015 MB	4095	16	63
ECHS translation	4032 MB	1024	128	63
Revised ECHS	7560 MB	1024	240	63
LBA translation	8032 MB	1024	255	63
16-bit cylinder	32255 MB	65535	16	63
28-bit CHS	130558 MB	65535	255	63
86Box	131071 MB	Not applicable		

The maximum supported disk image size for IDE or SCSI is 131071 MB. Disk overlay software such as *Ontrack Disk Manager* can work around BIOS limits and allow booting of IDE hard drives within the 131071 MB limit, with the same caveats as using such software on a real machine.

2.6.2 Floppy disk images

Supported formats:

Format	File extension
Raw image	Many *
<i>86F</i>	.86f
CopyQM	.cq / .cqm
DiskDupe	.ddi
EZ-DisKlone Plus	.fdf
Formatted Disk Image	.fdi
HxC MFM	.mfm
ImageDisk	.imd
Japanese FDI	.fdi
PCjs JSON	.json
Teledisk	.td0

* Raw image extensions recognized by 86Box include: .bin .dsk .flp .hdm .ima .img .vfd .xdf

Floppy disk detection

86Box determines the physical media format (sides, tracks per side, sectors per track, bytes per sector) of a floppy disk image through the following methods:

1. Image file header data - not applicable for **Raw** and **DiskDupe** formats;
2. DOS BIOS Parameter Block data within the image;
3. If all else fails, a guess is made based on the image file's size.

The BIOS Parameter Block detection method may behave incorrectly with non-DOS floppy disks. Installation floppies for UNIX and Linux are common examples of non-DOS disks. Disabling *Check BPB* is strongly recommended when accessing these, as an inaccurate BPB detection may result in read errors, data corruption and other issues.

Note: When using a **Raw** image of a non-DOS floppy with Check BPB disabled, make sure the image file is not truncated (smaller than its media size), otherwise incorrect behavior may still occur.

2.6.3 MO / ZIP removable disk images

Supported formats:

Format	File extension
Raw image	Many *
Japanese FDI	.mdi (MO)
	.zdi (ZIP)

* Raw image extensions recognized by 86Box include: .ima .img

2.6.4 CD-ROM / DVD-ROM optical disc images

Supported formats:

Format	File extension
Cue sheet	.cue + .bin
ISO	.iso

CD audio

Compact Disc Digital Audio (CDDA) playback through the emulated CD-ROM drives is supported on **Cue sheet** images. Audio output is enabled on the first CD-ROM drive and muted on subsequent drives by default; individual drives can be muted or unmuted through the *status bar* or *Media menu*.

Note: Only raw format (.bin) tracks are supported. Compressed or otherwise encapsulated audio tracks (.wav, .mp3, .ogg, .flac and other formats) are not supported.

2.6.5 Cassette tape images

Supported formats:

Format	File extension
Raw PCM audio	Many *
PCE cassette	.cas
Wave audio	.wav

* Raw audio extensions recognized by 86Box include: .pcm .raw

2.6.6 PCjr cartridge images

Supported formats:

Format	File extension
Raw image	Many *
JRipCart	.jrc

* Raw image extensions recognized by 86Box include: .a .b .bin

2.7 Tertiary and quaternary IDE

The additional tertiary and quaternary IDE controllers, enabled through the *Storage controllers* settings page, are not supported by all emulated BIOSes and may require manual configuration of emulated operating systems. The specific details are outlined on this page.

2.7.1 System resources

The following resources are used by these additional controllers:

Channel	Main I/O port	Status I/O port	IRQ
Tertiary	0168h	036Eh	10
Quaternary	01E8h	03EEh	11

Each controller's IRQ can be configured through its respective *Settings* button on *Tertiary / Quaternary IDE Controller*. The *Plug and Play* option on the *IRQ* box enables Plug and Play functionality, allowing a PnP compliant operating system to automatically set the controller's IRQ, while all other options set a static IRQ with no Plug and Play.

Note:

- When using a non-Plug and Play IDE controller on an emulated machine which supports Plug and Play, remember to mark the IRQ as being used by a legacy ISA device in the BIOS setup utility.
 - Many operating systems do not allow non-Plug and Play IDE controllers to use IRQs outside of the default ones listed on the table above.
-

2.7.2 BIOS support

The tertiary and quaternary controllers are not visible and not bootable by the BIOS on most machines currently emulated by 86Box, no matter whether or not they are Plug and Play.

Machines with **MR BIOS version 3** are the rare exception to this rule, since that BIOS provides full support for non-Plug and Play controllers (as long as the *default IRQs for each controller* are used), including bootability and INT 13h services.

2.7.3 Operating system support

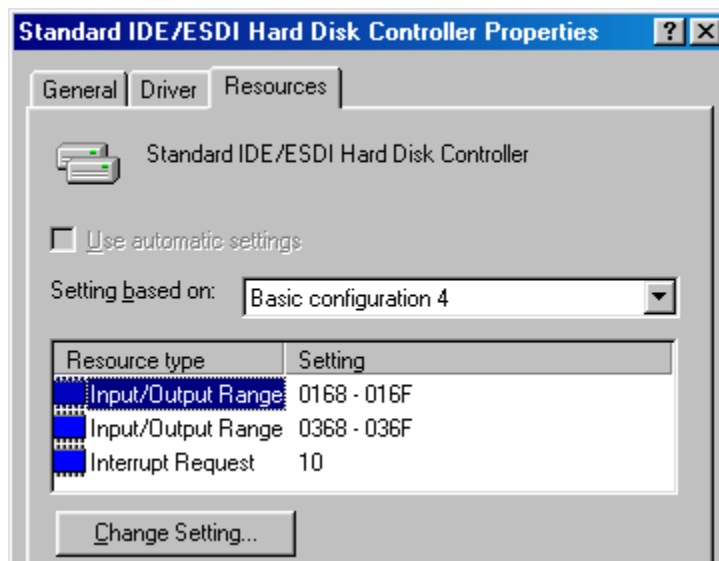
DOS and real mode

DOS and other real mode operating systems rely on INT 13h services provided by the BIOS to access hard disks. These are only provided for the tertiary and quaternary channels by **MR BIOS version 3**, as mentioned above.

Windows 95, 98 and Me

The **Windows 9x family** will automatically detect Plug and Play IDE controllers on boot. Non-Plug and Play controllers will be detected during installation *only if the BIOS supports them*. Follow these steps to enable a non-Plug and Play controller on an already-installed system:

1. Go to the *Add New Hardware* control panel.
2. Add a *Standard IDE/ESDI Hard Disk Controller* from the *Hard disk controllers* category.
3. Don't restart the system when asked to.
4. Go to the *Device Manager* tab of the *System* control panel.
5. Select the newly-added *Standard IDE/ESDI Hard Disk Controller* device from the *Hard disk controllers* category and click *Properties*.
6. Go to the *Resources* tab.
7. Select *Basic configuration 4* in the *Settings based on* box.
8. Change the resource settings to match the I/O ports on the [table above](#) and the configured IRQ. The first *Input/Output Range* range corresponds to the **main** I/O port, the second one corresponds to the **status** I/O port, and *Interrupt Request* corresponds to the IRQ.
 - The status I/O port range is off by 6. Select 0368 for the tertiary channel or 03E8 for the quaternary channel.
 - The screenshot below shows an example configuration for the tertiary channel.
9. If both the tertiary and quaternary channels are enabled, repeat the steps above to enable the other controller.



Windows NT, 2000 and XP

Windows 2000 and XP will automatically detect Plug and Play IDE controllers on boot. Additionally, **Windows NT 3.5, 4.0, 2000 and XP** will automatically detect non-Plug and Play controllers during installation, regardless of BIOS support; however, this auto-detection of non-PnP controllers does not work on most machines with **Award BIOS**.

Note: If you install the system to a hard disk on one of the additional controllers, it will not be bootable unless *the BIOS supports booting from these controllers*.

On **Windows 2000 only**, non-Plug and Play controllers can be enabled on an already-installed system through *Add New Hardware* similarly to *Windows 9x as shown above*. The resource parameters cannot be changed, and therefore, only the *default IRQs for each controller* are supported. *Basic configuration 0003* corresponds to the **tertiary** channel, while *Basic configuration 0002* corresponds to the **quaternary** channel.

Windows Vista and 7

The **Windows NT 6 family** does not support legacy (ISA or VLB) IDE controllers, and therefore cannot use the additional channels as currently emulated by 86Box.

Linux

There are different steps for enabling additional IDE controllers on Linux, depending on which IDE driver stack is used by your distribution's kernel.

Modules can be loaded at any time with the `modprobe` command, or loaded on boot by adding the module's name (and parameters if required) to a file in `/etc/modules-load.d` on newer systemd-based distributions, or the `/etc/modules` file on older distributions.

- **libATA** (typically kernels **2.6.19 and above**):
 - Load the `pata_isapnp` module to enable Plug and Play controllers.
 - Load the `pata_legacy` module with the `probe_all=1` parameter to automatically detect and enable non-Plug and Play controllers. Only the *default IRQs for each controller* are supported.
- **Legacy IDE** (typically kernels **older than 2.6.19**):
 - Load the `ide-pnp` module to enable Plug and Play controllers.
 - Non-Plug and Play controllers require editing the kernel command line on your bootloader to add each controller's I/O ports and IRQ:
 - * **Tertiary:** `ide2=0x168,0x36e,10` (assuming IRQ 10)
 - * **Quaternary:** `ide3=0x1e8,0x3ee,11` (assuming IRQ 11)

Note: Some distributions may automatically detect additional IDE controllers; however, that is very rarely the case.

2.8 Networking

86Box supports two connection modes for the *emulated network cards*. The specific details on these connection modes and network emulation as a whole are outlined on this page.

2.8.1 SLiRP

SLiRP creates a private network with a virtual router, allowing the emulated machine to reach the host, its network and the Internet; on the other hand, the host and other devices on its network cannot reach the emulated machine, unless *port forwarding* is configured. This is similar to the **NAT** mode on other emulators and virtualizers.

The virtual router provides automatic IP configuration to the emulated machine through DHCP. If that is not an option, use the following static IP settings:

- **IP address:** 10.0.2.15
- **Subnet mask:** 255.255.255.0
- **Default gateway:** 10.0.2.2
- **DNS server:** 10.0.2.3

The host can be reached through IP address 10.0.2.2, while other devices on the host's network can be reached through their normal IP addresses.

Note: SLiRP is only capable of routing TCP and UDP traffic. Other protocols such as IPX and NetBEUI can only be used with *PCap* networking.

2.8.2 PCap

PCap connects directly to one of the host's network adapters. The emulated machine must be configured as if it were a real machine on your network. This is similar to the **Bridge** mode on other emulators and virtualizers.

This mode requires *Npcap* (or another WinPcap-compatible driver) to be installed on the host. Only **wired Ethernet network connections** are compatible; Wi-Fi and other connections will not work at all, as they do not allow PCap to listen for packets bound to the emulated machine's MAC address.

Private PCap network

If you have an incompatible network connection on your host system (such as Wi-Fi), or if you wish to connect the emulated machine to the host without also connecting it to your network, a private network can be created with PCap in one of two ways:

- Install and configure the *Microsoft KM-TEST Loopback Adapter* included with Windows.
 - Guides on how to install this adapter are available online.
 - The adapter alone only provides a direct connection to the host, with no DHCP server, therefore requiring manual IP configuration on both the host and the emulated machine.
 - Windows' *Internet Connection Sharing* feature can be used to connect the emulated machine to the host's network and the Internet, with DHCP for automatic IP configuration, similarly to SLiRP but with the added benefit that the host can reach the emulated machine without port forwarding.
 - * Port forwarding can be performed through Internet Connection Sharing's *Settings*.

- If VMware is installed, use one of the VMnet adapters included with it.
 - *VMnet1* (Host-only) connects to the host only.
 - *VMnet8* (NAT) connects to the host, its network and the Internet.
 - * Port forwarding can be performed through the Virtual Network Editor's *NAT Settings*.

2.8.3 Advanced features

The following advanced features can be accessed by directly editing the emulated machine's configuration file, which is `86box.cfg` by default.

MAC address

With the exception of **[LPT] Parallel Port Internet Protocol**, every emulated network card stores its MAC address in the `mac` directive of its respective configuration file section. Only the suffix (last three octets) of the MAC address can be edited; the prefix (first three octets) will always be the card manufacturer's **Organizationally Unique Identifier**, such as `00:E0:4C` for Realtek.

Example: MAC address `00:E0:4C:35:F4:C2` for the Realtek RTL8029AS

```
[Realtek RTL8029AS]
mac = 35:f4:c2
```

SLiRP port forwarding

Port forwarding allows the host system and other devices on its network to access TCP and UDP servers running on the emulated machine. This feature is configured through the `[SLiRP Port Forwarding]` section of the configuration file.

Each port forward must be assigned a number, starting at 0 and counting up (skipping a number will result in all subsequent port forwards being ignored), which replaces `X` on the following directives:

- `X_protocol`: Port type: `tcp` or `udp` (default: `tcp`)
- `X_external`: Port number on the host (default: same port number as `X_internal`)
- `X_internal`: Port number on the emulated machine (default: same port number as `X_external`)

The host system can access forwarded ports through `127.0.0.1` or its own IP address, while other devices on the network can access them through the host's IP address.

Note: The emulated machine's IP address must be set to `10.0.2.15` (the default IP provided through DHCP) for port forwarding to work.

Example: forward host TCP port 8080 to emulated machine port 80, and host UDP port 5555 to emulated machine port 5555

```
[SLiRP Port Forwarding]
0_external = 8080
0_internal = 80
1_protocol = udp
1_external = 5555
```

2.9 ISABugger

The ISABugger card provides a debugging interface for software developers, consisting of two 8-bit hexadecimal displays and two banks of 8 LEDs, all controlled by the emulated machine. It can be enabled through the *Peripherals settings page*.

These displays and LEDs are shown on the *status bar* as described in the diagram below:



2.9.1 Background

From `src/device/bugger.c`:

Implementation of the ISA Bus (de)Bugger expansion card sold as a DIY kit in the late 1980's in The Netherlands. This card was a assemble-yourself 8bit ISA addon card for PC and AT systems that had several tools to aid in low-level debugging (mostly for faulty BIOSes, bootloaders and system kernels...)

The standard version had a total of 16 LEDs (8 RED, plus 8 GREEN), two 7-segment displays and one 8-position DIP switch block on board for use as debugging tools.

The "Plus" version, added an extra 2 7-segment displays, as well as a very simple RS-232 serial interface that could be used as a mini-console terminal.

2.9.2 Registers

The ISABugger's control registers can be accessed through the following operations on I/O ports 7Ah and 7Bh:

- **Writing:** write the register's index to port 7Ah, then write the value to port 7Bh.
- **Reading:** write the register's index to port 7Ah, then read the value from port 7Bh.
- **Index reading:** the last register index written to port 7Ah can be read back from the same port. The most significant bit is always set, as an indicator that the ISABugger is enabled.

Note: The ISABugger I/O ports only support byte (inb/outb) operations. Word (inw/outw) and dword (inl/outl) operations will result in undefined behavior; so will selecting or attempting to read back an unknown register index, or performing an illegal operation such as reading from a write-only register.

Register reference

Index 0x00 - Red LEDs (write-only)

Index 0x01 - Green LEDs (write-only)

Each LED bank shows a binary representation of the 8-bit value written to its register, from the most significant bit on the left to the least significant bit on the right. Setting a bit will light up its corresponding LED (displayed as **G** or **R**), and clearing a bit will dim its LED (displayed as **g** or **r**).

Index 0x02 - Right display (write-only)

Index 0x04 - Left display (write-only)

Each display shows a hexadecimal representation of the 8-bit value written to its register.

Index 0x20 - Serial port data (not implemented) (read/write)

Index 0x40 - Serial port configuration (not implemented) (read/write)

While the aforementioned real ISABugger card is equipped with an independent RS-232 serial interface, that feature is currently not implemented on 86Box in a user-facing manner.

Index 0x80 - Initialize (not implemented) (write-only)

This register has no effect on 86Box, as the emulated ISABugger is always enabled and ready.

Index 0xFF - Reset (special)

Writing register index 0xFF to port 7Ah will immediately reset all registers to their startup value, clearing all displays and LED banks.

This is a special register which cannot be read or written; writing to port 7Bh immediately after a reset will result in the value being sent to the default register index of 0x00, which corresponds to the red LEDs.

2.10 Build guide

Follow these steps to compile 86Box from source:

1. Install the [MSYS2](#) environment. The rest of the guide will refer to the directory that you install it to (C:\msys32 or C:\msys64 by default) as the MSYS2 root.
2. Launch your MSYS2 environment using the *MSYS2 MinGW 32-bit* shortcut. If you do not wish to use the shortcut, launch it using the *mingw32.exe* executable in the MSYS2 root.
3. **Once launched, you should update the environment:**

```
$ pacman -Syu
```

You may need to do this twice, just follow the on-screen instructions. Please re-run the command periodically to keep the environment up-to-date.

4. **Run the following command to install all of the dependencies:**

```
$ pacman -S gdb make git mingw-w64-i686-toolchain mingw-w64-i686-openal mingw-
↳w64-i686-freetype mingw-w64-i686-SDL2 mingw-w64-i686-zlib mingw-w64-i686-
↳libpng
```

5. **Once the environment is fully updated and all dependencies are installed, change directory to src:**

```
$ cd path/to/86Box/src
```

6. **Start the actual compilation process:**

```
$ make -f win/Makefile.mingw
```

`make` does not run in parallel by default. Use the `-j` switch to specify the number of parallel threads, such as `-j4` for 4 threads. You should not exceed your system's thread (logical processor) count, as that uses more resources for little to no gain.

8. If compilation succeeded (which it almost always should), the `86Box.exe` executable will be in the `src` directory.
9. In order to test your fresh build, replace the `86Box.exe` in your current 86Box environment with your freshly-built one. If you do not have a pre-existing 86Box environment, download the latest successful build from the [86Box Jenkins](#), and the latest ROM set from the [roms repository](#).
10. Enjoy using and testing the emulator! :)

If you encounter issues at any step or have additional questions, please join the IRC channel or the appropriate channel on our Discord server and wait patiently for someone to help you.

2.11 Advanced builds

The [86Box Jenkins](#) provides all kinds of pre-release testing builds for advanced users. These are linked to the [86Box git repository on GitHub](#); a new build is produced with the latest source code every time the repository is updated.

Important: Testing builds are development snapshots which may contain bugs, unfinished features or other issues. These should only be used if you know what you're doing.

2.11.1 Variants

86Box builds are available in a number of variants. The Jenkins page will automatically detect the recommended variant for the system you're viewing it on, but if you're downloading builds for a different system (or you have disabled JavaScript), use the guide below to select a variant:

- The **Old Recompiler** is recommended. The **New Recompiler** is in beta; you may experience bugs and performance loss with it.
 - The Old Recompiler is not available for the ARM architecture. You must select the New Recompiler for running 86Box on ARM Linux systems such as the Raspberry Pi.
- On **Windows, x86 (32-bit)** is recommended even if you have a 64-bit system.
 - x64 (64-bit) allows for emulating more than 2 GB of RAM on some later machines and using larger soundfonts with FluidSynth, at a slight performance loss.
- The regular variant (**86Box**) is recommended; it is compiled with the `--preset=regular` CMake flag, enabling the default feature set.
 - Release versions of 86Box are based on this variant.
- The debug variant (**86Box-Debug**) is compiled with the `--preset=debug` CMake flag, which provides debug symbols and no optimizations, to allow for debugging with `gdb` and other tools.
 - This variant runs slower than the standard one due to the removal of optimizations and addition of debugging features.

Discontinued variants

- Dev variants (**86Box-Dev** and **86Box-DevODR**) as of November 18th 2021.
 - These variants contained incomplete and experimental features subject to change at any time, with the -Dev variant also containing the New Recompiler beta.
- Optimized variants (**86Box-Optimized**) as of March 18th 2021.
 - These variants' aggressive microarchitecture-specific optimizations provided very little performance improvement (within margin of error on modern CPUs) while introducing bugs and other incorrect behavior.
 - Optimized binaries can still be produced by *compiling 86Box from source* with the `--preset=optimized` CMake flag, which enables optimizations for the build host's CPU microarchitecture. No support will be provided for those.

2.12 File formats

86Box introduces new file formats for disk images and other purposes. These formats are documented on this section.

2.12.1 86F

A floppy disk surface image format which stores data in FM- or MFM-encoded transitions.

Preliminary specification for v2.20

All offsets are in hexadecimal. This specification is subject to change before its final release.

```

00000000: Magic 4 bytes ("86BF")
00000004: Minor version (0x14)
00000005: Major version (0x02)
00000006: Disk flags (16-bit)
    Bit 0      Has surface description data (1 = yes, 0 = no)
                This data indicates if the corresponding bit on the FM/MFM_
↳ encoded surface                is a normal bit or a special bit (weak bit or hole, depending on_
↳ the other bit):                0 = The corresponding FM/MFM encoded surface bit is_
↳ normal                        1 = The corresponding FM/MFM encoded surface bit is_
↳ either a weak bit or a hole:   Corresponding FM/MFM encoded bit is 0:_
↳ Hole (noise on read, not overwritable) Corresponding FM/MFM encoded bit is 1:_
↳ Weak bit (noise on read, overwritable)
    Bits 2, 1  Hole (3 = ED + 2000 kbps, 2 = ED, 1 = HD, 0 = DD)
    Bit 3      Sides (1 = 2 sides, 0 = 1 side)
    Bit 4      Write protect (1 = yes, 0 = no)
    Bit 5      Bitcell mode (1 = Extra bitcells count specified after
                disk flags, 0 = No extra bitcells)
                The maximum number of extra bitcells is 1024 (which
                after decoding translates to 64 bytes)
    Bit 6      Revolutions (0 = one revolution, 1 track has 16-bit number of_
↳ revolutions)
00000008: Offsets of tracks
    Note that thick-track (eg. 360k) disks will have (tracks * 2) tracks, with each_
↳ pair of tracks                being identical to each other.
    Each side of each track is stored as its own track, in order (so, track 0 side 0,
↳ track 0 side 1,                track 1 side 0, track 1 side 0, etc.).
    The table of the offsets of tracks is 2048 bytes long, each track offset is an_
↳ unsigned 32-bit                integer. An offset of 00000000 indicates the track is not present in the file.
    As an example, an 86F representing a disk with 80 thin tracks and 2 sides per_
↳ track, where all                the tracks are present in the file, would have the first 160 offsets filled in,_
↳ same for a disk

```

(continues on next page)

(continued from previous page)

with 40 thick tracks and 2 sides. Same with only 1 side but only the offsets at
 ↪00000000, 00000008,
 etc. (so every second offset) would be filled in.

Track offset + 00000000: Track flags (16-bit)

Bits 4, 3 Encoding

00 = FM

01 = MFM

10 = M2FM

11 = GCR

Bits 2, 1, 0 Bit rate, if encoding is MFM:

000 = 500 kbps

001 = 300 kbps

010 = 250 kbps

011 = 1000 kbps

101 = 2000 kbps

If encoding is FM, the bit rate is half that.

The RPM is determined from track length and data rate.

Track offset + 00000002: Total bit cells count (32-bit)

Track offset + 00000006: Bit cell where index hole is (32-bit)

Track offset + 0000000A: FM/MFM/M2FM/GCR-encoded data (track length bytes)

Track offset + 0000000A + track length: Surface description data if present (track
 ↪length bytes)

If this is a multi-revolution 86F, then track offset + 00000000 has a 16-bit number of ↪

↪track revolutions,

and the track header + data appears for each revolution, while surface description data, ↪

↪if present,

can appear any number of times, but only once per encoding + bit rate combination.

This needs work to properly make surface data work with flexible multi-revolution ↪

↪support.

Track lengths:

The total bit cells count is always present.

The track is stored as (bit cells >> 8) bytes, with one extra bit cells if the ↪

↪number of bit cells

is not divisible by 8.

2.12.2 Extended HDI (HDX)

A derivative of the *Japanese FDI* disk image format, with a more compact header as well as support for images larger than 4 GB.

Specification

All offsets are in hexadecimal. The [Translation] values are for future use.

00000000:	Signature (59 54 44 44 20 A8 78 D7 / "YTDD " A8 78 D7)
00000008:	Full size of the data in bytes (64-bit)
00000010:	Sector size in bytes (32-bit)
00000014:	Sectors per cylinder (32-bit)
00000018:	Heads per cylinder (32-bit)
0000001C:	Cylinders (32-bit)
00000020:	[Translation] Sectors per cylinder (32-bit)
00000024:	[Translation] Heads per cylinder (32-bit)
00000028:	Raw data (size set in offset 00000008)