

WinFellow

User Manual

TORSTEN ENDERLING (torsten.enderling@gmail.com)
WORFJE (worfje@gmx.net)

petschau.github.io/WinFellow

v0.5.11 (2024/03/21)

Contents

I	Disclaimer	4
II	GNU General Public License, Version 2	4
III	About WinFellow	5
IV	Release Notes	6
V	System Requirements	7
VI	Essential Requirements	8
VII	First Steps	11
VIII	Frequently Asked Questions	17
1	Miscellaneous	17
1.1	I can't get <put your favorite game or demo here> working.	17

1.2	Where can I find/download the Kickstart ROM file?	17
1.3	I installed Workbench, but it refuses to load up.	18
1.4	The keys O and P aren't working, why is this?	18
1.5	How to play games and run applications with WinFellow?	18
2	Performance	18
2.1	Sound playback is choppy when the emulation window is in the background	18
3	Floppy disk images	18
3.1	Which file formats are supported?	19
3.1.1	ADF images	19
3.1.2	ADZ and DMS images	19
3.1.3	IPF images	19
3.2	Saving games doesn't work.	19
4	Filesystems and hardfiles	20
4.1	What are hardfiles?	20
4.2	How do I use a hardfile?	20
4.3	Can I use my existing UAE hardfiles (*.hdf) with WinFellow?	21
5	Filesystems	21
5.1	What are filesystems?	21
5.2	How do I use a filesystem?	22
6	GUI	23
6.1	How can I play a game that has more than one disk / How can I switch a disk(image)?	23
7	Known Bugs	23
7.1	The player is invisible in <insert your favourite game here>	23
7.2	The player doesn't collide with other game objects	23
8	I still have questions!	23
IX	Module Ripper	23
9	The Module Ripper	24
9.1	Credits	24
9.2	Supported Module Formats	24
9.3	How to invoke the mod-ripper?	24
9.4	File naming conventions	25
9.5	Dump Chipmem option	25
9.6	Final note	25

X	Contributing to WinFellow	26
XI	Credits	26

Part I

Disclaimer

WinFellow is provided without warranty of any kind.

None of the authors of WinFellow shall in any event be liable for any direct, consequential, incidental, indirect or special damages even if advised of the possibility of such damages. In particular, the authors shall have no liability for any damage loss or corruption of data or programs stored in or used in conjunction with WinFellow, nor shall the authors be liable for the cost of retrieving or replacing damaged lost or corrupted data.

Part II

GNU General Public License, Version 2

WinFellow is developed and distributed under the terms of the GNU General Public License, Version 2. For the exact terms of the license, please see the file gpl-2.0.pdf, that should have been distributed together with this manual.

Part III

About WinFellow

Welcome to the WinFellow Amiga Emulator, a software emulation of the Commodore Amiga computer system. WinFellow provides a virtual Amiga environment to your PC in which you can run Amiga software. The emulator runs on an IBM compatible PC with a Windows operation system installed. Different Windows versions can be used, the emulator has been tested successfully on operating systems ranging from Windows 7 to Windows 10.

The Commodore Amiga 500 model was used as a reference Amiga system when creating WinFellow, and the default emulator configuration aims at providing the speed and feeling of this particular Amiga system to your PC.

WinFellow is also highly configurable, advanced features can be enabled to provide more powerful capabilities that surpass those of a stock Amiga 500. It is strongly advised to read the feature summary so you can get familiar with the strengths and limitations of the emulator.

The Amiga 500 computer was first released in 1987 as the successor for Amiga 1000. It served for many years as the entry-level and the most popular member of the Amiga computer system family. With its rich multimedia capabilities thanks to several co-processors devoted to the generation of fast graphics as well as 4 channel sample-based stereo sound, it soon became a popular platform for games and demos. The advanced multimedia capabilities combined with a multi-tasking, window-based operating system made the Amiga well ahead of its time for a home computer.

WinFellow was created by Petter Schau; it is based on the Fellow Amiga Emulator for DOS, of which the first public version was released in December 1996. Leading up to the release there were several years of curious experimentation with methods to do fast emulation of selected parts in the Amiga hardware. Today it is a mostly complete software emulation of an Amiga 500 and beyond, thanks to the contributions of several other people.

Part IV

Release Notes

What's New in WinFellow 0.5.11?

Thank you for your interest in WinFellow. Compared to the earlier version 0.5.10, the following changes are included in this build:

- New features:
 - the default display driver was changed from DirectDraw to Direct3D 11, where supported
 - implemented support for keyboard-initiated reset in Amiga Forever
- Bug fixes:
 - improved logging in case of failure to initialize Direct3d graphics driver
 - fix several issues related to creating, loading, applying and saving configuration files
 - fix potential memory leaks
 - implement more failure checks for Direct3D initialization
 - fix crash to desktop when using an emulated joystick in Amiga Forever
- Maintenance updates:
 - code refactoring:
 - * added sound driver interface and classes
 - * redesign filesystem wrapper and logging
 - * upgrade to cpp20
 - removed a Windows XP specific workaround from the filesystem module
 - reformatted the project using a uniform clang-format configuration

Please see the included file ChangeLog.txt for a full list of changes, including minor bug fixes which may not be listed above.

Part V

System Requirements

WinFellow is highly optimized for execution on low-end systems. It can be executed on any of the following operating systems:

- Windows 7
- Windows 8 (including Windows 8.1)
- Windows 10

It does not have any particularly high CPU requirements, and should work well on any system capable of running one of these operating systems. However, for advanced features like 2x graphics mode, a relatively fast CPU will be required (a minimum of 2GHz is recommended).

Graphics card requirements

DirectDraw

Support for DirectDraw rendering is the default configuration, and is supported on all operating systems. For optimum performance, the graphics card should support DirectDraw hardware acceleration. Use is possible without, but performance will be reduced drastically.

Direct3D 11

Support for Direct3D 11 can be enabled in the display configuration settings, which enhances performance on systems that support it. It requires DirectX 11 to be installed, which is available for Windows 7, 8, 8.1 and 10.

Part VI

Essential Requirements

Before you can begin to use WinFellow, you will need two things: an Amiga Kickstart ROM, and a floppy image.

1. ROM image

A copy of an Amiga Kickstart ROM, which is taken from a REAL Amiga. The Kickstart is copyrighted software and is not included with the emulator. This file can be extracted using a program called TransROM, which is included with WinFellow (Utilities\Amiga).

The correct way to obtain this file is to copy it from the Amiga you own; you can also buy Amiga Forever from Cloanto which contains licensed Kickstart ROM images and Workbench software prepared for immediate use with the emulator.

Using TransROM

TransROM must be copied over to a real Amiga in order to extract a copy of the Kickstart ROM chip and capture it into a file.

Example

```
CLI> transrom >RAM:kick.rom
```

Will copy the Kickstart ROM chip and write it into a file in RAM: called "kick.rom". That file must then be copied to the PC and into the WinFellow directory (although it may be called from any different directory).

2. Floppy image(s)

You will also need an "image" of a real Amiga floppy disk, containing the program which you wish to run under WinFellow. This is obtained using a program called "TransDisk", and it is also included with WinFellow.

A disk-image is a file in which the data normally stored on a floppy-disk is collected. A PC's floppy drive cannot read an Amiga floppy disk, so to be able to use WinFellow, you NEED a disk image of a real Amiga disk.

Using TransDisk

Like TransROM, TransDisk must be copied over a real Amiga to be of any use. TransDisk is run from the AmigaDOS shell prompt, and will copy a floppy disk, track by track, into a file (which can be read by WinFellow).

Example

```
CLI> transdisk >RAM:df0.adf
```

Will make an image of any disk present in DF0:. The disk doesn't have to be named "df0.adf", you can use whichever name you feel is appropriate.

3. Copying Files From PC to Amiga & Amiga to PC

There are different options to transfer files between your Amiga and your PC. This list is by no means meant to be exhaustive, it just provides a few different options. With a little research, other ways may be possible or more fitting for your particular use case. Cloanto also kindly provides a few options here.

(a) Transfer vial terminal package and null-modem cable

If your PC is still equipped with a serial port, you can utilize a null-modem cable to perform a Zmodem transfer between an Amiga and a PC. Simply plug the cable into a serial port on each machine, load up a terminal package such as Terminus, NComm or Term on the Amiga, or HyperTerminal or an alternative like RealTerm on the PC, and initiate a Zmodem upload on the Amiga. If you have it configured correctly (automatic ZModem download at the receiving end should be enabled), the PC will automatically accept this upload, and will download the software onto your PC's hard disk.

For best results, obtain a copy of BaudBandit serial.device replacement for the Amiga, and set both ends to 57,600 Baud. BaudBandit should allow transfers of up to 5500cps to/from a 68000-based Amiga - it is available from AmiNET.

If you're one of those people with an AmigaOS 1.2/1.3 A500, with only one floppy drive, you could be in trouble. Unless you have a Terminal package on a floppy disk which you can boot or use in a second drive, there's no way you can use a null-modem cable to talk to the PC (which is essential to actually get software like TransDisk or TransROM onto an Amiga floppy). Your best bet would be to approach an Amiga user group, who may be able to supply you with such a disk, or if you're lucky, an Amiga shareware house/distributor in your area. Note: Some Amigas permit you to lock the serial port at 115,200 baud. This largely depends on the machine, it's motherboard revision, and it's processor. By locking the serial port at 115,200 baud on both the Amiga and the PC, you can effectively transfer files over a null-modem cable at over 10,100cps.

(b) CrossDOS

If you don't have a null-modem cable, or don't particularly want to buy one, you can transfer the Amiga software over the PC using a 720k PC floppy disk. Workbench 2.1+ come bundled with CrossDOS, which is software which allows you to read PC 720k disks in your Amiga's floppy drive. If you don't have CrossDOS (i.e., you don't have access to

Workbench 2.1 or higher), you can also use MessyDOS which is located at on Aminet. Here's a step by step guide on how to transfer software by floppy:

- i. Obtain a 720k PC diskette. Format it either on your PC, or on the Amiga using CrossDOS.
- ii. Create either an .ADF file or Kickstart ROM image using the method described above.
- iii. Archive this file using either LhA or Zip (Make sure you have these on the PC, so you can unarchive them when you take the disk over).
- iv. Copy the archive to the 720k diskette, using CrossDOS or MessyDOS.
- v. Take the diskette over to the PC, copy the archive to the hard-disk, and unarchive it. Copy the Kickstart ROM image or .ADF file into the WinFellow directory and you're ready to go!

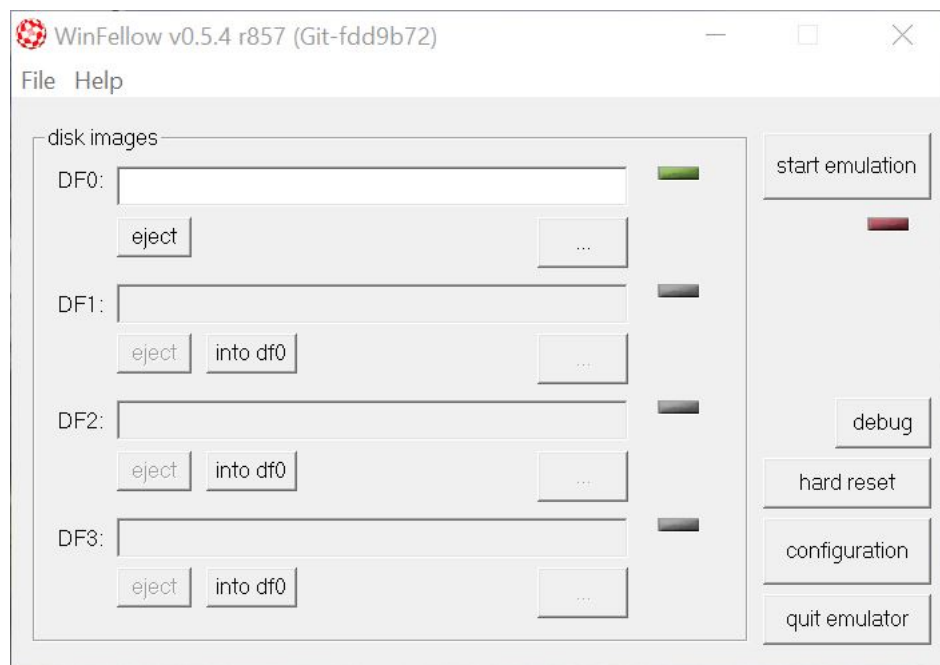
(c) Amiga Explorer

Cloanto provides a shareware tool called Amiga Explorer that can also assist in the process, which provides more options to connect the systems; for example, it can be used with a Bluetooth serial adapter. Amiga Explorer is also included in Amiga Forever.

Part VII

First Steps

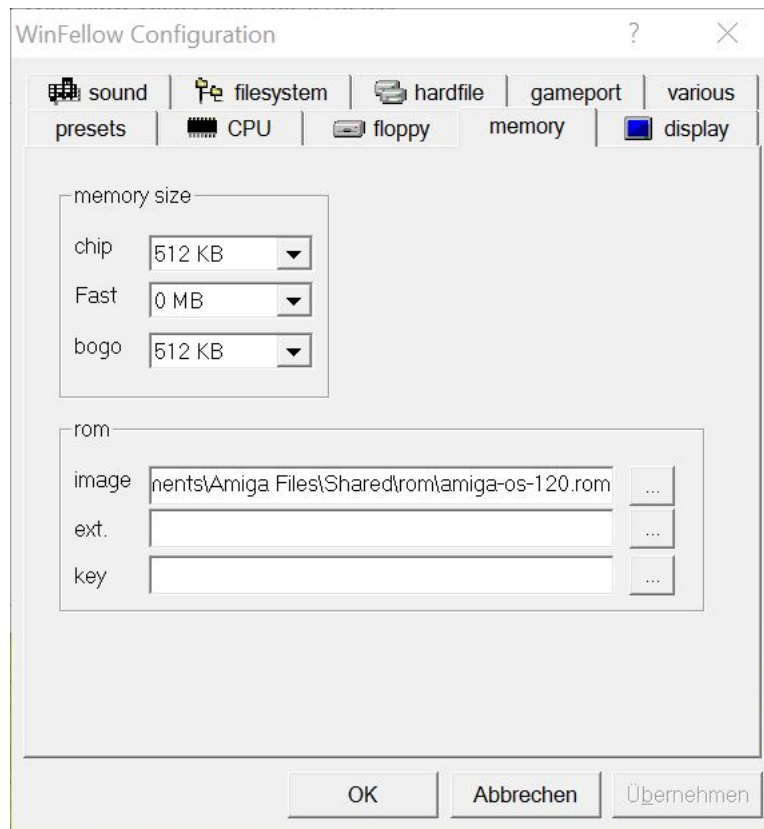
This guide is going to take you through the first basic steps required to configure an emulation session using a ROM image and a boot floppy, showing you the most important configuration settings that you should be familiar with along the way.



Above screenshot shows WinFellow's main menu. To get started, a number of settings should be configured.

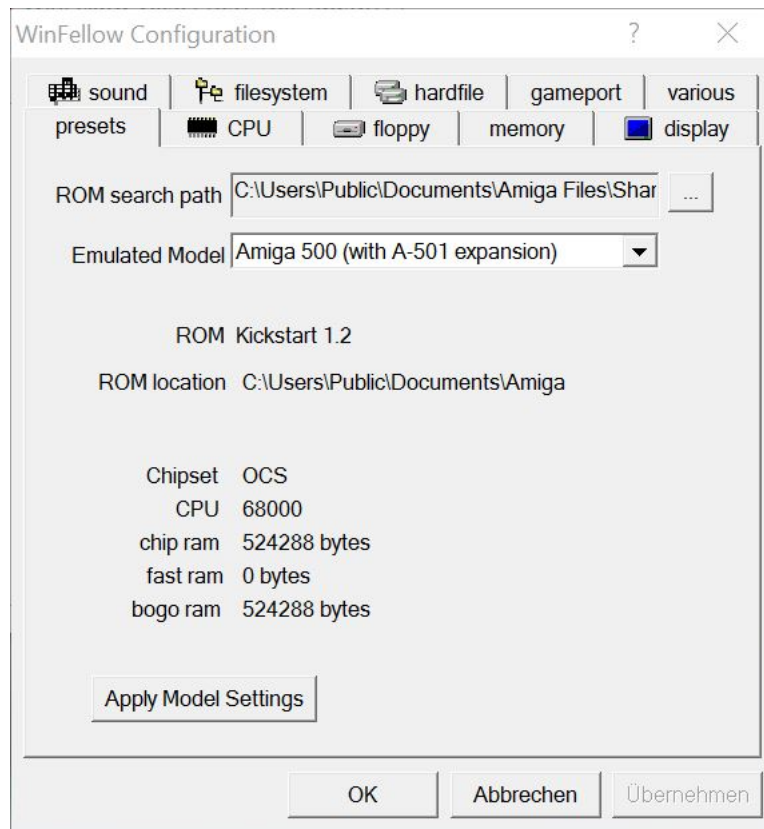
You can quickly select the disk images that are used in floppy drive DF0: to DF3: on this screen. Only floppy drives that are enabled in the configuration menu can be configured here; the green LED besides DF0: indicates that the drive is enabled. The red LED below the "start emulation" button is the power led; the dark red tone in the screenshot it indicates that no emulation session is active.

Configuration of ROM As a first step, you should configure the ROM image that you want to use. To configure the ROM, you should click the "configuration" button, switch to the "memory" tab and configure a ROM, like in the example below:



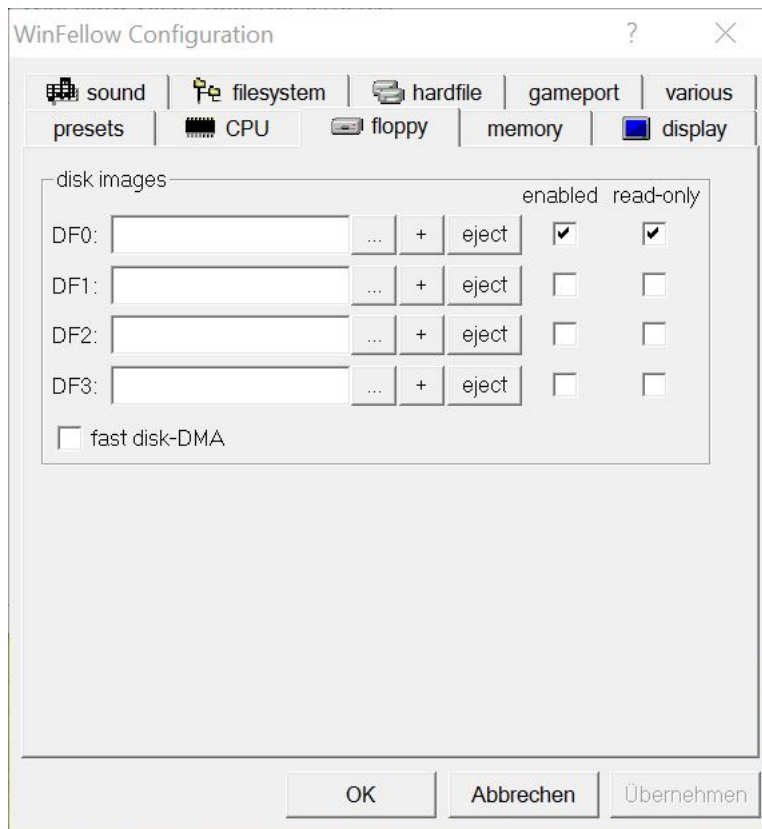
Click the “...” button besides image to open a dialog that allows you to select the base ROM image you have stored earlier; confirm that dialog by hitting OK. While on the “memory” tab, review the settings configured under “memory size”. It is currently configured for 512KB chip memory size, as well as 512 KB bogo memory size, which is a pretty common setup. You can change these values individually, if you know that the software you are going to use has special memory requirements.

Configuration of presets If you want to emulate a system based on a particular Amiga model, a quick way to get started is to utilize the “presets” tab:



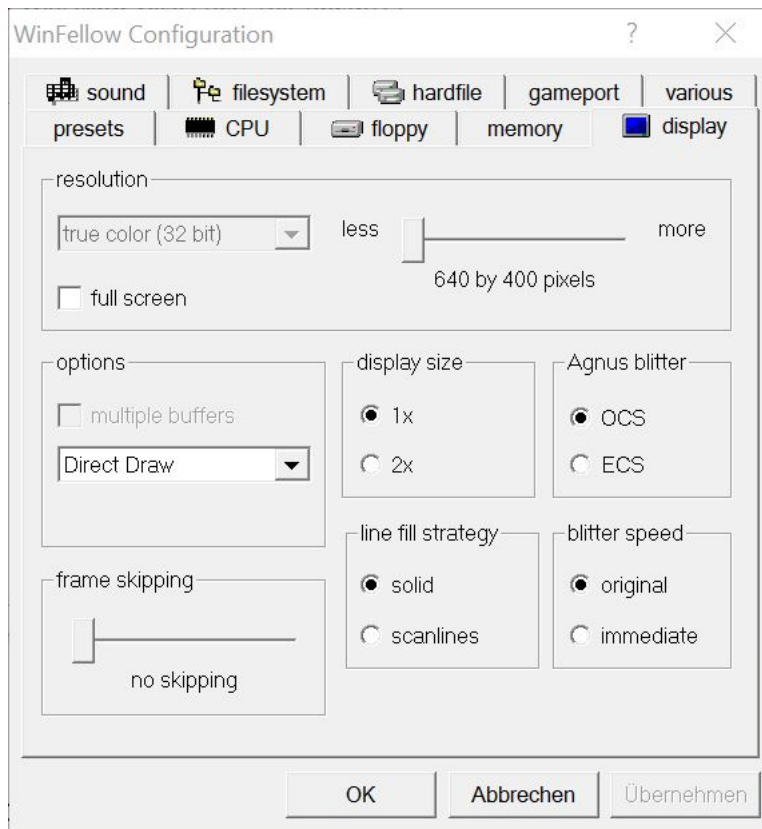
You first configure a ROM search path where WinFellow will automatically attempt to locate a matching ROM. You can then select a configuration from the dropdown list of emulated models; for example, you might select the Amiga 500 model with the popular A-501 expansion like shown above. After selecting a model, the most important settings for that model are shown in a preview below. WinFellow will attempt to locate the correct ROM for your selection automatically. You can apply the settings shown in the preview to the active configuration by clicking “Apply Model Settings”.

Configuration of floppy disks Switch to the “floppy” tab to configure the floppy disk images used by the emulation session:



You can configure the image that is used by each of the four drives (DF0:-DF3:) in this dialog. Each drive can be enabled and disabled individually, and the disk image can be configured as read-only. Enabling “fast disk-DMA” can speed up the access to the emulated floppy disks. This setting however is not compatible with all software, you should test if the software you want to use supports it or leave it disabled.

Configuration of display settings While still in the configuration menu, switch to the “display” tab:



Review the display settings configured here. You can adjust the resolution of the emulation window to your preference by moving the “resolution” slider. You can also enable full screen mode on this tab and configure the screen resolution. Depending on your screen size, it may make sense to configure the display size to 2x; you should adjust the resolution accordingly. When you are done, confirm your changed configuration settings by clicking the OK button. This brings you back to the main menu.

Input device focus There are a few important key definitions you should be aware of when using a WinFellow emulation session. You will notice that the mouse cursor will disappear when you start the emulation session, and keyboard input will not appear on the Windows screen. This is because the emulation session takes control of your input devices, so that keyboard, mouse and possibly a joystick or gamepad are redirected to the emulated system. You can pause an emulation session and return to the main menu by pressing the *F11* key. You can toggle input device focus between an emulation session and Windows by pressing the *F12* key repeatedly; to give the focus back to the emulation session, ensure that the window is in the foreground and the mouse cursor over it when pressing *F12*.

Starting and stopping an emulation session Click the button “start emulation” to start an emulation session. An additional window will appear hosting the emulation session; the virtual environment is now running. Go back to the main menu by pressing *F11*.

Saving the configuration Then, when in the main menu, open the “File” menu, select the option to “Save Configuration”. You might also save the configuration under a certain name by selection “Save Configuration as...”. WinFellow will load the configuration that was used last automatically when you start it again later. The most recently used configurations are also listed in the “File” menu. When you try to quit the emulator, and there are unsaved configuration settings, a warning message will appear to remind you that you might want to save your settings.

Closing the emulator You can close the emulator by clicking “quit emulator” in the main menu.

Part VIII

Frequently Asked Questions

1 Miscellaneous

1.1 I can't get <put your favorite game or demo here> working.

Many older games require special settings. Generally if having trouble check the following check-list:

- some Kickstart versions require special CPU models, so play around with reasonable combinations of them; for example it is a bad idea to run Kickstart 1.2 on a '030 CPU
- some games are written for a fixed amount of Bogo or Chip memory, so try setting them both to 512 kB or even disable Bogo memory completely
- not all games behave nicely (some control the hardware directly) so that you may run into trouble when using things like "Fast Disk-DMA" or the immediate blitter
- many old games only make use of the first floppy drive and some do the strangest things when you try to use two drives
- timing may be critical, as some games do timing delays not by the timer but by delay loops; this will result in trouble when using CPU speeds other than the original 7 MHz
- If in doubt, use a default A500 setting like:
 - 68000 CPU at 7 MHz
 - 1 floppy drive (disable the others!), leave "Fast Disk-DMA" unchecked
 - 512 kB Chip and eventually 512 kB Bogo RAM
 - Kickstart 1.3 or 1.2
 - normal, delayed OCS blitter
 - no filesystems or hardfiles, "Disable Autoconfig-devices"

If this configuration doesn't work and it's really an A500 game (not an AGA thing or such) it's possibly a WinFellow problem.

1.2 Where can I find/download the Kickstart ROM file?

You can buy the Kickstart ROM at Cloanto.

1.3 I installed Workbench, but it refuses to load up.

Though it is possible to install Workbench into a directory, you must make sure that you install it into your hardfile or filesystem root directory.

1.4 The keys O and P aren't working, why is this?

Since the 'enable autofire' is placed on those keys, the keys aren't used as input for the Amiga keybuffer. Just remove Keyboard Layout 1 from gameport 1 or 2 and the P and O keys will work again.

1.5 How to play games and run applications with WinFellow?

This is the general procedure to play a game or run applications on WinFellow:

1. start WinFellow
2. click on the configuration button
3. select the memory property sheet
4. select a KickROM in the kickstart image edit control
5. click the OK button to return to the main window
6. select a ADF image containing your game
7. start the emulation by clicking start emulation

That's it, you should see the emulation window booting the KickROM (switching from dark gray, light gray to white) and then loading from the disk.

2 Performance

2.1 Sound playback is choppy when the emulation window is in the background

Many people like to run the emulator in the background to be able to listen to music. Should you encounter stuttering playback in Direct3D graphics mode, check your graphics card's driver settings and ensure that there is no frame limit configured for background applications.

For nVidia GPUs: the nVidia Control Panel contains a setting called "Background Application Max Frame Rate", which defaults to 30 frames per second. Increase the value to 50, or disable the limit to have proper background audio playback.

3 Floppy disk images

Floppy disk images are used to simulate floppy drives connected to the emulated Amiga. This is necessary, as a PC can't read Amiga floppy disks.

3.1 Which file formats are supported?

WinFellow supports *.adf disk images and the compressed *.adz, *.adf.gz or *.dms images; support for these is built into WinFellow, so you don't need any additional executables.

3.1.1 ADF images

Normal *.adf files are similar to a hardfile; they contain the contents of an Amiga floppy disk dumped into a file and thus are always 880 kb large for normal DD floppies. WinFellow can handle these without problems.

Note that there is also an "Extended" ADF file format, which is used for example by the game backups you can download from the Factor 5 homepage; these files contain some tracks which are specially encoded; you can recognize these easily by their file size, as they are always larger than 880 kB. WinFellow has limited support for these, and they worked fine in our tests. Should you experience any crashes if you attempt to use them, please let us know!

3.1.2 ADZ and DMS images

ADZ and DMS images may be regarded as normal ADF files that have been compressed to reduce their size; when inserting such files WinFellow decompresses them into a temporary ADF file, which is then used for your emulation session. Note that any changes saved to compressed images will be lost as soon as you quit the emulator. So to write changes to a disk decompress it manually before starting to work on it.

3.1.3 IPF images

Experimental support for *.ipf files (floppy disk images generated by the Software Preservation Society) is available, but will only work in a few cases (the image must not be "flakey", which usually means copy protected). IPF images will only function when the user DLL from the IPF Support Library is downloaded and copied into the WinFellow directory. The Windows version 5.1 has been tested to work.

3.2 Saving games doesn't work.

When trying to save games there are some things you should take care of:

- make sure the savedisk file is not write protected (when copied from a CD-ROM to your HD the file usually is write protected)
- the file must not be compressed (see section about floppy image file formats); when saving to a compressed image, the changes are only saved to the uncompressed temporary image which becomes deleted when you quit the emulator

- take care of the compatibility information described in section 1.1; often savedisk problems go away when using a more compatible setting as disk access is always a time critical thing
- some games require savedisks to be formatted in a special way; sometimes a freshly formatted empty AmigaDOS floppy will do, sometimes you have to create a disk in the game; if possible, try using the savedisk that comes with the game or obtain a working one

4 Filesystems and hardfiles

4.1 What are hardfiles?

Hardfiles are used to simulate a harddrive connected to the emulated Amiga. There are two types of hardfiles:

1. The first type is most often just referred to as “hardfile” or “plain hardfile”, and it is technically a 1:1 copy of the contents of an Amiga hard disk partition into a file. Plain hardfiles can be booted from using Kickstart 2.0 or higher. This device supports only a subset of the features a real Amiga hard drive has (you can for example defragment it).
2. The second type is called “RDB hardfile”, and it is a hardfile that simulates the copy of a full hard drive which has been prepared with a rigid disk block using tools like HD Toolbox. This device supports all features a real Amiga hard drive has (you may for example create partitions, or defrag it). It can contain multiple partitions, and can under some conditions be booted from using Kickstart versions as early as 1.3.

A plain hardfile can be converted into an RDB hardfile by creating a rigid disk block using HD Toolbox.

Both types of hardfile are of a fixed size and don't grow as they are written to (in contrast to, for example, the Amiga RAM disk, or the filesystems that are explained below). When creating a new hardfile you have to specify the size the hardfile shall have, i.e. the amount of storage space you will be able to use on the simulated Amiga hard drive.

4.2 How do I use a hardfile?

First of all you need a hardfile that is to be used. Perhaps you already have one from UAE or DOSFellow; if you don't have one you can create an empty one in the configuration menu under the hardfile tab. Note that a newly created hardfile is just a file containing scratch; you need to format it on the Amiga side to be able to write any data to it (plain hardfiles), or prepare it using HD Toolbox (RDB based hardfiles).

Add the hardfile to your config in the configuration menu's hardfile tab. Make sure that "Disable Autoconfig-devices" (section Various) is **not** checked. The

Autoconfig functionality is needed for the Amiga to automatically detect and initialize the device.

If you're going to use an already existing file (and this file is bootable) eject all inserted floppy images and boot from it (you should make sure the used Kickstart version fits the one the hardfile has been installed with).

The minimum Kickstart version required to boot from a hardfile depends on the type of hardfile.

"Plain" hardfiles require Kickstart version 2.0 or newer to boot from them; with Kickstart 1.x the only way to use these hardfiles is by the use of mount files and booting directly from them is not possible.

RDB based hardfiles can be booted from using Kickstart 1.3 if FastFileSystem is installed in the RDB; from version 2.0 onwards FastFileSystem is contained in the ROM.

If you want to prepare an RDB based hardfile using HD Toolbox, make sure to use the device with SCSI_DEVICE_NAME "fhfile.device". It can be configured in the properties of the HD Toolbox icon, or you can call HD Toolbox in a CLI window and pass the device name fhfile.device to it.

There are plenty of tutorials available online that explain this process, which is not specific to WinFellow; make sure to take the following points into account:

1. WinFellow's SCSI-based hardfile device name is fhfile.device
2. the emulated Amiga must be rebooted after an RDB has been created using HD Toolbox; WinFellow's hard reset function can be used to achieve that

An example of a guide that can be followed is linked to below:

[How to use HDToolbox on Workbench 3.x](#)

4.3 Can I use my existing UAE hardfiles (*.hdf) with WinFellow?

Yes, WinFellow and UAE use the same hardfile format. When using hardfiles generated by UAE, these must not make use of the Picasso96 uaegfx-driver, as this one is not yet supported by WinFellow. It also must not make use of AGA screenmodes.

5 Filesystems

5.1 What are filesystems?

Like a hardfile, a filesystem is also used to simulate a storage device connected to the emulated Amiga. But instead of using a file you configure WinFellow to use a directory on one your windows devices that shall be used to store or read data.

This has the advantage that you don't have to care about the amount of data you want to store.

But this also has disadvantages: though on the first look the simulated device looks like a harddisk, it misses some features: formatting or defragging a filesystem

device is absolutely impossible (and would not make sense either). To defrag it you'll have to run the windows defrag over the harddrive you stored the files on.

Another disadvantage lies in the differences between the Windows and the Amiga filesystem. The Amiga filesystem makes use of 8 different storage flags (e.g. executable, writable, ...). The Windows filesystem only supports 3 flags, but even these aren't really usable for this purpose so that actually only the windows archive flag is used to represent Amiga flags. For an Amiga, filenames like " " or ".. " are fine; Windows runs into some serious trouble when you try to use such names. Just try to create a file named CON, then you'll know what I mean... ;)

To get rid of this problem, a technique called fsdb (FileSystem DataBase) has been introduced in UAE (and WinFellow, as WinFellow uses UAE code for filesystem support).

Every time a file has a name not supported by Windows, or uses flags that can't be stored, or has a comment, an entry for this file is created in a special database file. This file stores the local name, the according Amiga filename, the optional comment and the flags that are used. That way it is possible to use the device exactly the same way an Amiga device could be used, allowing to even boot from a filesystem (this was impossible without the fsdb).

Why is all this relevant?

WinUAE uses a slightly different way of storing the flag information, as it also utilizes the write-protection bit to represent the Amiga's writable and deletable flags; we do not think that this is the optimal solution, and thus implemented it another way. This means that the filesystem support from WinUAE is not necessarily 100% compatible with WinFellow's support. *So be careful when exchanging data over a filesystem between them both.*

5.2 How do I use a filesystem?

Configure the filesystem in the configuration menu on tab filesystem. You might also check "Automount Windows-drives"; this will automatically mount all hard disks, connected network drives and CD-ROMs as Amiga devices (so you don't have to configure anything manually).

Make sure that "Disable Autoconfig-devices" (section Various) is *not* checked. The Autoconfig functionality is needed for the Amiga to automatically detect and initialize the device.

As with hardfiles, ensure that you use a Kickstart version 2.0 or greater.

Now you need something you may boot from. I suggest a Workbench disk or a hardfile. The device will appear under the name you configured. Install Workbench to the filesystem if you like and try booting without the floppy inserted.

6 GUI

6.1 How can I play a game that has more than one disk / How can I switch a disk(image)?

You can switch disks(images) at runtime in WinFellow. You can do this by pressing the function key <F11> during the emulation. After pressing the key you can see the main window again and this way you can change the disk(images). By clicking the "Start Emulation" button you can continue the emulation at the point before you pressed the function key.

7 Known Bugs

7.1 The player is invisible in <insert your favourite game here>

There are certain games in which sprites aren't rendered correctly, resulting in invisible objects. Known games having this problem are Stardust, SuperFrog, Robocop 2, Zool, Alien Breed and Fire And Ice.

7.2 The player doesn't collide with other game objects

Not all collision types are properly implemented in WinFellow; for example in Transplant the spacecraft never collides with the rocks flying around. Known to have this problem is Transplant, but there probably are much more titles having this problem.

8 I still have questions!

If you have questions not covered in this manual, find problems not mentioned here or simply don't understand something written here, don't hesitate to mail one of us. This manual is a work in progress, and to be able to maintain it we need your input. Also, feel free to post a question to the WinFellow support forum.

Part IX

Module Ripper

Since many users asked about it the, old module ripper contained in the DOS-based Fellow has been ported into the new WinFellow environment and has undergone some major changes. This document is meant to describe WinFellow's mod-ripper in it's current form.

9 The Module Ripper

9.1 Credits

Special thanks must go out to ExoticA for the excellent file format section, and Sylvain "Asle" Chipaux for his Amiga Mod Packers Described Page.

Without the information provided on these pages the mod-ripper wouldn't have been possible. Another source of information was the Extended Module Playersource code, which also has some helpful information about various module formats.

9.2 Supported Module Formats

The following formats are currently supported; more are probably to come later (we'll have to see the demand):

- ProTracker and clones (Protracker, Noisetracker, Startrekker 4/8 channel, ...)
- SoundFX 1.3 and 2.0
- SoundMon 1.0, 2.0 and 2.2
- FredEditor
- ProRunner 2.0
- ThePlayer 4

Note that while ProTracker files play fine in many music replayers like WinAmp, this isn't the case for the other more exotic formats. When you look for a good module player capable of playing all kinds of Amiga formats try XMPlay with the DeliX plug-in. DeliPlayer is an option as well, but is no longer under active development.

9.3 How to invoke the mod-ripper?

The mod-ripper is a bit hidden in Fellow's integrated debugger. When running it, you are first prompted if you want to scan the emulated Amiga's memory for modules. This is the recommended way to rip modules; start the game or demo you want to rip the music from and when the song you want to rip is actually playing you may halt the emulation and run the mod-ripper to gather the song out of the memory.

The other possible way is to rip from floppies; the module ripper detects when floppies are inserted and prompts you for each one if you want to rip from that floppy; this works ok with some games or demos that use custom trackloaders but will usually result in damaged modules when used on formatted AmigaDOS disks so that you have to be a bit careful with modules ripped off a disk.

9.4 File naming conventions

Whereas common module types like Protracker have a unique ending (*.mod), this is not the case for many other formats. For these, the ripper uses the default extension .cus to indicate a format that is not very widespread; these are the ones that won't play in most players so you'll need either a capable Amiga module player or a converter to play these.

9.5 Dump Chipmem option

In the debugger you also find a button "Dump Chip-memory". This may be used to save the contents of the whole chipmem into a file and so allows you to run other external module rippers over that.

9.6 Final note

The module ripper is in an experimental state. Should you experience issues with an exported file, send WinFellow's logfile along with a description of what you tried to rip music from (game/demo name) and the information the file save requester showed to Torsten Enderling so that we can fix the problem.

Part X

Contributing to WinFellow

WinFellow is developed by a team of volunteers in their spare time; you are very welcome to make contributions of your own. Contributions can be improvements to the code, new features, but can also include documentation updates or whatever else you feel could be an improvement. It is best if you find something you want to work on rather than mailing us and ask to be assigned a task. Since WinFellow is developed on an idealistic basis, the basic idea is that everyone should do something they like.

When you start doing something it is wise to mail some of us in advance to make sure no one is doing double work. Since WinFellow is developed as open source software hosted on GitHub, it is easy to contribute. You can fork the source code of WinFellow, create a branch for your topic and start committing changes to it; your changes can be submitted to our project in form of a pull request. Raising a pull request early on allows us to discuss your request early, so that we can see if/how your request may fit into WinFellow before you put any effort into it.

There is a how-to guide in Markdown syntax within the source code archive in the directory “fellow\doxygen\Documentation” called “HOWTO Development environment setup.md”, that can help you get started setting up a development environment for WinFellow.

Feel free to mail us (petschau@gmail.com / torsten.enderling@gmail.com) if you are interested in helping!

Part XI

Credits

The WinFellow Team

Created and developed by

PETTER SCHAU (petschau@gmail.com)

with help from

TORSTEN ENDERLING (torsten.enderling@gmail.com)

The Beta Tester Team

Special thanks go out to the beta tester team. Through their dedication, dozens of bugs could be fixed before they were ever released to the public.

BRIAN ANDREASEN

GIUSEPPE BAGNATO

BENNI P

ATTILA BORSANYI
JAN BROCKMANN
MATTHEW BUSHELL
JIM DURDAN
ANDREAS EIBACH
DAMIEN GREEN
VINCENT HAOT
MARCO SASCHA SVEN HOPPE
PAUL IRVINE
MALC JENNINGS (ale@emuhq.com)
MAGNUS JENSEN
NICK K. KALOGERAS
NICK LAMBURN
MARCELO NOGUEIRA MAGRI
TARNYKO RASTY
ANDY RILEY
DOMINIK SCHMIDT
BRYN THOMPSON
TOMASZ TOMASZEK
PHIL UNDERHILL
ROBERT VAN BUREN
CHRISTOS VASDEKIS
ANDY VOSS
TIM WILSON aka Emulator Chicken

Programming Contributions

WORFJE (worfje@gmx.net)

- cross-platform development
- graphics module
- sound module
- GUI
- many changes over the years...

TEDMAUL (dan@dillinja.org)

- partially porting Fellow to Linux

MND (novamarco@hotmail.com)

- win32 keyboard, joystick & mouse modules

ROMAN DOLEJSI (roman@sorry.vse.cz) & DAVID VORACEK (assassin@paru.cas.cz)

- various improvements in GUI
- keyboard LED code
- main design and partial implementation of 68010/20/30 instruction set emulation
- makeadf (David)
- generalized BMP dump code + various other improvements

RAINER SINSCH (sinsh@stud.uni-frankfurt.de)

- MOD-ripper
- Low-pass filter
- Wav-capture patch

Thanks Also Go To

ANDREAS AXELSSON, BRETT EDEN, HETZ BEN HAMO, MASASHI YAMAGUCHI, JEREMY CRAWFORD and the DOS Documentation Crew: ROBERTO GASPARRINI, LAURENT DUGOIS, ANDREAS EIBACH, LOU C. FER, KENNETH VESTERGAARD SCHMIDT, PAWEL JEDRAK, PETER STOLK and MARCELO GURGACZ, as well as numerous others for taking interest in the emulator.

We'd also like to thank KELVIN SHERLOCK for providing various bugfixes around CPU emulation.

Special thanks go out to the Cloanto RetroPlatform team (MIKE C. BATTILANA, OSCAR SILLANI and anyone else who may have been involved) for their support integrating WinFellow into Amiga Forever, and for their continuing efforts.

Thanks also go out to Toni Wilen for making his great cputester tool available for public use.

Website Translations

Mnd (Italian)
 Wizard (German)
 Sha! (French)
 Bachfire (Danish)
 Petter (Norwegian)
 Apple (Finnish)
 Solamnic (Greek)
 Leon (Turkish)
 SewerRat (Portuguese)
 Szatan (Polish)
 Bigma (Dutch)
 OrB (Spanish)

Libraries used

zlib

Copyright © 1995-1998 JEAN-LOUP GAILLY and MARK ADLER

Other sources used

UAE Filesystem handler

Copyright © 1996-2000 BERND SCHMIDT, ED HANWAY, GUSTAVO GOEDERT and other UAE contributors. UAE filesystem sources are under the GNU GPL; used with permission.

xDMS

The xDMS v1.3 portable DMS archive unpacker is public domain software written by ANDRE RODRIGUES DE LA ROCHA used with his kind permission.