

# WinFellow User Guide

TORSTEN ENDERLING (carfesh@gmx.net)  
WORFJE (worfje@gmx.net)

v0.5.3 (2015/11/04)

## Contents

<b>I</b>	<b>Disclaimer</b>	<b>3</b>
<b>II</b>	<b>Credits</b>	<b>3</b>
<b>III</b>	<b>What's New in WinFellow 0.5.3?</b>	<b>6</b>
<b>IV</b>	<b>System Requirements</b>	<b>7</b>
<b>V</b>	<b>Frequently Asked Questions</b>	<b>7</b>
<b>1</b>	<b>Miscellaneous</b>	<b>8</b>
1.1	I can't get <put your favorite game or demo here> working. . . . .	8
1.2	Where can I find/download the Kickstart ROM file? . . . . .	8
1.3	I installed Workbench, but it refuses to load up. . . . .	9
1.4	The keys O and P aren't working, why is this? . . . . .	9
1.5	How to play games and run applications with WinFellow? . . . . .	9
<b>2</b>	<b>Floppy disk images</b>	<b>9</b>
2.1	Which file formats are supported? . . . . .	9
2.1.1	ADF images . . . . .	10
2.1.2	ADZ and DMS images . . . . .	10
2.1.3	IPF images . . . . .	10
2.2	Saving games doesn't work. . . . .	10

<b>3</b>	<b>Filesystems and hardfiles</b>	<b>11</b>
3.1	What are hardfiles? . . . . .	11
3.2	How do I use a hardfile? . . . . .	11
3.3	Can I use my existing UAE hardfiles (*.hdf) with WinFellow? . . . . .	11
<b>4</b>	<b>Filesystems</b>	<b>12</b>
4.1	What are filesystems? . . . . .	12
4.2	How do I use a filesystem? . . . . .	12
<b>5</b>	<b>GUI</b>	<b>13</b>
5.1	How can I play a game that has more than one disk / How can I switch a disk(image)? . . . . .	13
<b>6</b>	<b>Known Bugs</b>	<b>13</b>
6.1	The player is invisible in <insert your favourite game here> . . . . .	13
6.2	The player doesn't collide with other game objects . . . . .	13
<b>7</b>	<b>I still have questions!</b>	<b>13</b>
<b>VI</b>	<b>Module Ripper</b>	<b>13</b>
<b>8</b>	<b>The Module Ripper</b>	<b>14</b>
8.1	Credits . . . . .	14
8.2	Supported Module Formats . . . . .	14
8.3	How to invoke the mod-ripper? . . . . .	14
8.4	File naming conventions . . . . .	15
8.5	Dump Chipmem option . . . . .	15
8.6	Final note . . . . .	15
<b>VII</b>	<b>GNU General Public License, Version 2</b>	<b>16</b>

## 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

# Credits

### The WinFellow Team

Created and developed by

PETTER SCHAU (petschau@gmail.com)

with help from

WORFJE (worfje@gmx.net)

TORSTEN ENDERLING (carfesh@gmx.net)

### 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

BENNIP

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

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.

Thanks also 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.

## 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.

## Part III

# What's New in WinFellow 0.5.3?

Thanks for your interest in WinFellow. Compared to version 0.5.2 there are a number of changes in this build, with the most significant ones being:

- New features:
  - support for high DPI configurations, also when used together with Amiga Forever 2016
  - support for the original Amiga 1000 boot process from Kickstart disk boot; configure the A1000 bootstrap ROM like any other ROM image for the feature to be automatically enabled
  - support for keyboard-initiated reset - press Ctrl+Left Windows+Right Windows or Ctrl+Left Windows+Context Menu (as not all keyboards have a right Windows key) to initiate a reboot within an emulated session
  - support for extended ROM images; this gives the ability to use the AROS boot ROM
  - support for screenshots, clipping editor/live preview and scanlines in Amiga Forever
- Changed functionality:
  - WinFellow is now compiled using Visual Studio 2015
  - the default configuration contains 512kB chip/bogo memory now and has the real-time clock active (Amiga 500 with A-501 expansion)
  - the default configuration is now windowed and has LEDs active
  - chip memory mirroring has been implemented for OCS chipsets; the change was required for compatibility reasons, as some titles rely on it
    - \* because of this change, chip memory is now limited to 512kB when an OCS blitter has been configured; if you need to use more, please use the ECS Agnus blitter in your display configuration (it is an actual limitation of the OCS chipset)
  - 2x mode in Amiga Forever now scales via DirectX instead of the internal draw modules (design change was necessary to enable live preview in 2x mode)
- Bug fixes:
  - implemented bogo memory mirroring for ECS chipset with 512kB chip/bogo memory (required by the demo Move any Mountain)
  - fix for CIA timer logic (fixes the game Atomix)

- fixes for several CPU instructions:
  - \* '020 bit-field instructions were re-implemented (fix for Lotus Esprit Turbo Challenge 96k)
  - \* ASL instruction's overflow detection was improved
  - \* MULU flag check improved
- escape key special handling in Amiga Forever improved (some titles would not detect an escape key press before)

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

## Part IV

# System Requirements

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

- Windows XP
- Windows Vista
- 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).

For optimum performance, the graphics card should support DirectDraw hardware acceleration. Use is possible without, but performance will be reduced drastically.

## Part V

# 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 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.

### **2.1 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.

### 2.1.1 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!

### 2.1.2 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.

### 2.1.3 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 (32-bit) version 4.2 has been tested to work.

## 2.2 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

## 3 Filesystems and hardfiles

### 3.1 What are hardfiles?

Most of you probably already know this and may skip this paragraph. For the ones that are new to emulation here a short description:

A hardfile is used to simulate a harddrive connected to the emulated Amiga. Technically a hardfile is a 1:1 copy of the contents of an Amiga harddisk partition into a file. Having this in mind the restriction a hardfile has comes clear: a hardfile is of a fixed size and doesn't grow as it is written to (like perhaps the Amiga RAM disk). When creating a 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 harddrive. This device supports all features a real Amiga harddrive has (so you may for example defrag it).

### 3.2 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 may now create it 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.

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.

Also make sure you use a Kickstart version 2.0 or newer; with Kickstart 1.x the only way to use hardfiles is by the use of mount files (this possibility may be explained in more detail in a later version of this manual) and booting directly from them is not possible (since the old Kickstart versions don't support Autoconfig).

You should now be able to use the file. If you have configured a newly created file, you should insert a Workbench disk file fitting the configured Kickstart, boot it and format the hardfile the way you want it (it will appear as FELLOW0).

If you're going to use an already existing file (and this file is bootable, not only a data partition) eject all inserted floppy images and boot from it (again you should make sure the used Kickstart version fits the one the hardfile has been installed with).

### 3.3 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.

## 4 Filesystems

### 4.1 What are filesystems?

Like a hardfile, a filesystem is also used to simulate some kind of 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.*

### 4.2 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.

## 5 GUI

### 5.1 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.

## 6 Known Bugs

### 6.1 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.

### 6.2 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.

## 7 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 VI

# Module Ripper

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

## 8 The Module Ripper

### 8.1 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.

### 8.2 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, Noisetraacker, 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 is well, but is no longer under active development.

### 8.3 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.

## **8.4 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.

## **8.5 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.

## **8.6 Final note**

The module ripper is in an experimental state. Should you experience issues with an exported file, send Fellow'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 VII

# 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.