JOTD HD Install Collection

	COLLABORATORS		
	TITLE : JOTD HD Install Collection		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 27, 2022	

	REVISION HISTORY		
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

JOTD HD Install Collection

1.1   JOTD Startup documentation

```
        ********************************************************************

                        JOTD Startup V3.2b
            © Copyright 1995-2000 Jean-François Fabre/Ralf Huvendiek

********************************************************************

              Introduction

              Arguments/Tooltypes

              JST Degrader

              CD32 Joypad emulation

              WHDLoad compatibility

              HRTMon compatibility

              OSEmu - OS emulation

              Snapshots

              TroubleShooting

              Register

              Getting original games

              Disclaimer

              Acknowledgements

              FAQ
```

                          Author

                          Developpement team

                          History

1.2 Introduction

As you could notice, I created and uploaded many HD-Install & fix programs for
non-dos games, but all of them were standalone, and in case of an error in the
general loader routines, I had to remake the loader (relink) and upload again.

Bert Jahn had developped WHDLoad, which solves this problem because it allows to
load small, game-specific loaders and run them using its own degrade/hd/dos code.
This technique allows easy updates in case of bugs or new functionnalities.

That's what I decided to do myself. JST was born from a 8-hour adaptation of
my link library. It's an equivalent to WHDLoad but a bit more powerful,
as the system can be called in the slave loader, messages can be printed,
a gui can be opened, there are lots of cool functions, and much more.
It also allows easier programming of loaders (but WHDLoad is still cool!),
and can run normal DOS programs in degraded mode (68060 support).

Now JST is also able to iconify some loaders and also to snapshot
games to disk to return to level 34, where the tentacle robo monster
is really impossible to beat without cheating.

The limited version can only create snapshots (for bug reports), but the RESUME ←
   feature is
reserved to registered users only, of course.

The keyfile must be called jst.key and be located in the L: directory

JOTD HD Startup is © 1995-1999 Jean-François FABRE
JOTDStartup is © 1999 Jean-François FABRE

1.3 Getting orignal games

As I say in this documentation, doing patches for cracked software does not ←
   interest me,
because it's not very rewarding and it means lots of extra work.

I understand original software for the amiga is hard to find those days, and that' ←
   s why
I offer my services to help you:

- in case your original disks are damaged. In this case, send me the original ←
   disks which are
damaged (as a proof, and I'll also try to rewrite on them) along with some money ←
   for the return
of your disks (and more if you want :)). Of course contact me before sending your ←
    originals,

as I may not have the disks, and I may not able to recreate them from disk images. ←
    In that case,
I'll backup the installed version and will send it either on disks or by e-mail.

- in case you're looking for an original copy of a game. I know some 2nd hand ←
   stores where there
are some cool titles. Of course I don't sell copies or cracks!!! It's only second ←
   hand original
software with box and manual. Contact me for any request about that.

1.4 Registering JST

JOTDStartup is SHAREWARE!

From V0.3 I decided to ask a fee of US$10, 20DM, 5£ or 50FF to be able to become
a registered user. However, I will register whoever I think he or she did ←
   something
cool for me (already sent disks, money, sports cars, Sandra Bullock clones...).
Lots of people are already on the list. If you think I forgot you, contact me.

To register:

1) Fill in the registration form provided in the package and send it to me by e- ←
   mail
   (if you don't have e-mail, don't fill the form, I'll manage with your snail ←
      mail
    address to generate the keyfile)

2) Send the money to:

Jean-François Fabre
19 Rue Emile Duploye
34500 BEZIERS
FRANCE

Specify your e-mail or send a blank disk with the money (for non-internet users),
and I'll send you your personal keyfile for JST (+the latest installs for the
ones who sent a floppy but specify which ones in priority as the whole archive is ←
   big)

IMPORTANT: The only proper way to register is to send cash, French chèques, or ←
   eurochèques.
I recieved UK cheques and Postal Money Orders. They are national only and I could ←
   not cash
them. Too bad there's no euro currency yet (don't talk to me about cheques in euro ←
   ! :( )

Contact me if you've got any doubt about the way to send me money.

If you still think this is too expensive, you can use the "limited" shareware ←
   version.
It has no limitations compared to older free releases but the new features will
be reserved to registered version. However, if I fix a bug, it will be fixed in
both registered and shareware version of course.

All the future loaders will be compatible with the limited version.

I hope you understand JST and all the loaders were hard work for us, that's why
I decided it. If you can afford, please register, else, you can use the limited
version freely. I won't bug you (promised :))
Of course, registered users have a higher priority for e-mail support from me, ←
    even
if I don't always answer if I don't think it's interesting to do so (for you and ←
    for me).

For the recall, here's the list of features reserved for registered users:

- Snapshot reload (RESUME)
- Possibility to save games in a different directory than the game directory (←
    SAVEDIR)
- Possibility to load game data in a different directory than the loader (LOADDIR)
- Reload battery backed-up clock on exit
- Quiet mode (QUIET)
- Configurable quit key (QUITKEY tooltype/argument) when the loader supports it
- In some loaders, only the registered version allows to quit (sorry for that!)
- A clear conscience because they registered :)

Note to people who don't have internet access: If you don't receive your JST ←
    keyfile
two months after you sent the money, please write me back! I'm so untidy I could
have forgot to register you! No shame :)
And when you register, please specify the loaders/fixes you'd like, because I may ←
    have
them in store...

1.5 Arguments/Tooltypes

                    You can select the options:

- From the CLI by typing the arguments (JST ? for help)
- From the WB by activating/disabling the tooltypes

Some have effect only on loaders that support them. They will be noted are * ←
    OPTIONAL*
Some have effect on all loaders. They're noted as *GENERAL*

Here's a short description of the options.

- OBJNAME="object": lets you specify an object name (icon launch).

- NTSC (GENERAL): default for loader is PAL. NTSC changes this to NTSC (really?)

- TRAINER (OPTIONAL): activates trainer when there is one. It can be infinite ←
    lives, time, etc...

- QUIET (GENERAL): displays nothing and opens no window. No error messages, and no ←
    author message.
  Be sure that everything works OK before selecting this one.

- NOCACHES/NOCACHE (GENERAL): Sets CACR to 0. All caches are disabled. 68060 ←
    supported.

- LEAVECACHES (GENERAL): Don't touch the caches. Disabled by NOCACHES

- NOFAST (GENERAL): Always allocates extension memory
  from chip memory. If you got crap sprites, this could help. USeless in the case ←
      of 512K mem games.

- HDLOAD (GENERAL): to save some memory, you can load from HD instead of caching ←
    all the files in RAM.
  It works for diskfiles and normal files.
  Useless in games with only 1 diskfile, use LOWMEM instead.

- RESUME (GENERAL): resume a previously saved snapshot. This option uses absolute ←
    allocation so
  you may need to reboot to get some free memory at the required location. You can ←
      also specify
  NOFAST in the JST tooltypes to make your snapshots compatible from one memory ←
      config to another.

- EXECUTE (DOS GAMES): this tooltype will allow you to use JST as a normal ←
    degrader like
  KillAGA, but with 68060 support and more control over the system. See
                  JST degrader
                    for further information.

- FAKEPAD (DOS/CD32 GAMES): this tooltype allows to emulate the CD32 joypad with ←
    joystick
  and keyboard, and make believe to some CD32 games that you really own a joypad.
  Check the
                  joypad emulation section
                   for more details.
  FAKEPAD=1 will use the 1st keyset, FAKEPAD=2, will use the second, and so on.
  There are 3 different keysets.

- USERDATA: if the user passes a string to this argument/tooltype an wants to run ←
    a program
  using EXECUTE (degrader features only), then this string hold the arguments of ←
      the program.
  If it's a loader, it can be used to specify some parameters to the loader. ←
      Useful for disk
  rippers to pass track start/end. Max length allowed is 80 characters.

- LOWMEM (OPTIONAL): will not cache diskfiles in memory. Read directly from disk. ←
    Useful if you've
  got a A1200/020/NoFast. With normal files, LOWMEM consumes even less memory than ←
      HDLOAD. Some
  loaders may bring up and error message or crash with an exception.
  In that case they don't support this option yet :)

- NOOSSWAP (OPTIONAL): Disables in game HD loads/saves. Depending on the loaders, ←
    NOOSSWAP
  will try to access the floppy when you try to save/load a game, or will do ←
      nothing.

If you don't select NOOSSWAP, if the game has some data to write, it will do it ←
   on your
hard drive. It's rather safe (I've been using it for a while now) but you should ←
    avoid
filesystems or disk caches that perform copyback caching (AFS, possible in ←
   Dynamicache).
This is not generally harmful, except if you quit by a reset or the game locks. ←
    In that
case, your hard drive can be unvalidated.

- NOQUIT (only in degrader mode (EXECUTE)): some games contain autodetachable code ←
    and run
in background. If you try to run such a game with a degrader, the degrader will ←
    enhance
display/caches when the program autodetaches and *crash*. NOQUIT will cause JST ←
    to run
in degraded mode forever (until you reboot).

- DELAY (GENERAL): Waits a little while after every OS swap.
This can be useful for low speed media. It allows the drive to stop and leaves ←
    time for
interrupts to be processed before the system is frozen again (allows lots of ←
    games to
run from CD on a CD32).

- SAVEDIR (OPTIONAL): allows the user to specify a directory where the game data ←
   like hiscores
or savegames will be saved/loaded. Some loaders with HD saves don't support it ←
    yet.
This function is only available to registered users.

- LOADDIR (GENERAL): allows the user to specify a directory where the game files ←
   or diskfiles
should be loaded from. For registered users only.

- QUITKEY (OPTIONAL): enter here the raw keycode of the wished exit key. If the ←
   loader supports
it, this key is now the quit key. (ex: QUITKEY=$5F sets HELP as the quit key)
This function is only available to registered users.

- DEADLY (GENERAL): "Clean" installs should run with DEADLY on, because it trashes ←
    ExecBase
and redirects interrupt vectors to an error message, so no ROM code is called if ←
    an
unexpected interrupt occurs. Leaving Execbase alive can be unreliable and can ←
    trigger 81000005
GURUs in the case the loader still uses system routines such as AllocMem()
(which is not recommended, but it's up to the programmer. I generally remove all ←
    system
code in the game).

- NOVBRMOVE (GENERAL,68010+ only): do not use a VBR table to pre-handle the ←
   interrupts. Sets VBR
to zero, so the game interrupts are alone. This mode allows to run some games ←
    which would not
work without it (in that case, the NOVBRMOVE tooltype is set in the game icon in ←
    the install pack)

Some options are here for debug and developpement purpose only:

- DEBUG (GENERAL): useful only for bug reports. Makes a memory image on exit.
- LEAVEVBR (GENERAL): useful only for debugging purposes, and with a debugger such ←
    as Action-Replay,
         HRTMon (v2.17 or lower!) or ThrillKill. Don't select it unless you want ←
              to debug. Works only with EXECUTE
- TEST (GENERAL): The loader will not start. Only the options selected will be ←
   displayed.
- VERBOSE (GENERAL): The loader will display information while initializing (←
   memory, files...)
- DEADLY (GENERAL): Your loaders should all run with this one on
- FREEZEKEY (GENERAL): This tooltype allows to specify the key used to enter in ←
   the debugger (HRTMon)
  if one is installed. Same usage as QUITKEY.
- FREEZERMB (GENERAL): This tooltype allows to enter into HRTMon with the right ←
   mouse button.
  Useful when the keyboard does not respond.

- MMU (GENERAL): Three modes are available:

    NONE: This option disables *ALL* MMU handling. Set it if you experiment ←
       strange
    behaviour like corrupt GFX or freezes at startup.

    LAXIST: This option does not exits with an error if an illegal memory access
    occurs. Useful with loaders which did not fix all problems, but dangerous.

    FULL: This option traps illegal memory accesses (read and write). This option
    is very useful to bust 24bit access bugs or
    other weird coding used by poor hacker-like/geeky/pizza eating A500 ←
       programmers.

  MMU default value is FULL

Some options can be set by an environment variable:
Those are FREEZEKEY and SAVEDIR (ex: setenv FREEZEKEY $5C, setenv SAVEDIR RAD:)

EXAMPLES from CLI for normal JST objects:

JST MagicPocketsHD NOCACHES DELAY NTSC

JST HookHD NOFAST HDLOAD SAVEDIR=ram: (registered users only can use SAVEDIR)

Examples from CLI for normal DOS executables:

JST Dune NOCACHES EXECUTE (the game does not like caches)

JST ssboot EXECUTE USERDATA=SWOS (the ssboot executable needs SWOS as an argument)

JST AGADemo LEAVEVBR LEAVECACHES NTSC EXECUTE (the demo likes caches and does not ←
   care about the VBR
relocated in fastmem, but I've got a multiscan WB, and I want to boot in 15KHz ←
   with NTSC)

Of course if the dos executable sets PAL or opens an intuition screen, display ←
    features of JST
will be cancelled. JST's no David Copperfield.

1.6 Start and end scripts

Registered version only!

With the registered version, you're able to make JST run scripts on startup and
when you quit the loader. Useful to disable/enable commodities, ethernet cards,
enforcer, etc...
Those 2 scripts are not specific to each loader but general. They must be located
in S: and be called:

jstpre.sh for the pre-JST script
jstpost.sh for the post-JST script

jstpre.sh will for instance contain:

enforcer off
handlecx multicx disable

This feature is also active (and very useful) with EXECUTE option (degrader mode).
You won't have to make a script for every game you install.

NOTE: if T: does not exist, the scripts won't be executed (PLEASE keep T: in RAM: ←
    or
a virtual disk)

1.7 Snapshots handling

Registered version only!

When a loader supports iconify, the iconify menu will allow you to snapshot the ←
    game data.
The data will be saved in the SAVEDIR directory if set, or in the game directory.

To load a previously saved snapshot, set the RESUME option.

If the loader or JST changes, your snapshots are no longer valid. This is a ←
    technical limitation
I cannot remove. So be careful before upgrading to a new JST or a new loader.

Only some loaders support the Iconify/Snapshot option at the moment. Every loader ←
    has to be
slightly modified (means modify/test/update) for this purpose. Updating every ←
    loader could
take time, you see...

1.8 TroubleShooting

Some error messages can occur before or after the game has started:

- Object is not relocatable: the game-specific loader must not address variables ←
   in an
  absolute way. If you wanted to run a DOS program using JST, use the EXECUTE ←
     tooltype.

- Invalid Object: someone tried to load non-loader objects or other files. To run ←
   a normal DOS
  program with JST don't forget to set EXECUTE tooltype/argument.

- Not enough memory: try to free some memory. Playing with NOFAST and HDLOAD and ←
   boot without
  startup-sequence can help too.

- Version of JOTDStartup is outdated for this object: get a newer version of JST, ←
   this one
  is just too old for the loader and would not work.

- Unsupported version: some games exist in different versions. I try to support ←
   some of them,
  but if the program does not recognize the version, it can display this message (←
     it can crash too!).

- Exception: <cryptic message> : some illegal instruction was encountered while ←
   the game executes.
  Try NOCACHES and NOFAST to see if it helps. Else report me/patch author the ←
     error with accuracy.

- Uninitialized level <X> interrupt: remove the DEADLY mode :)

- Wrong virtual disk ID: patch error. Contact author.

- Diskfile not found, Disk Read Error: in HDLOAD mode, a file is missing.

- Snapshot is not compatible with this version: you made a snapshot with an old ←
   loader,
  an old JST or on another amiga. You cannot resume your game!

- GetDiskPointer called while in LOWMEM mode: LOWMEM is not available with this ←
   loader
  (rare message)

- Not enough memory while you've got plenty: check the priority of your different
  memory expansion cards, because JST uses MEMF_REVERSE, and maybe that's the ←
     cause. Use the
  VERBOSE option to see what's the problem.

- OpenFakeExec: feature removed, ask for an update: the loader uses OpenFakeExec
  whereas I removed this feature because OSEmu is much more powerful. The loader
  needs to be updated. Write me.

If the game simply hangs or resets, check that you don't have some MMU related ←
    program
running (enforcer, VMM...) or low level patches like Fastexec. My patches are ←
    reported
to work with softkicked amigas, though.

Some versions of the 68060.library could bring up problems because they remap ←
    addresses
from $0 to $1000 in fastmem, so copper and blitter are lost. From JST 2.0, MMU is ←
    fully
managed (thanks to Ralf!), unless MMU is set to NONE.

The game can crash if it finds an exception and there is not enough memory to ←
    restore the
system and display the exception message. Buy some more mem :-)

Some old versions of JST may be able to run a loader while the newest cannot. In ←
    that case,
please report me the problem urgently.

I'm currently able to test JST/my loaders on the following processors/amigas

A500/68000/2Megs (small game like POP)
A1200/68030/18Megs (the one I've just bought)
A1200/68060/48Megs (the one I'm developing with)
A1200/68040/34Megs (the one from a friend)

I ain't got access to a PPC card, but maybe I'll buy one in the future for my ←
    A1200.

1.9 FAQ

Q: The loader exits with some message like: Version problem

A: Your version is not supported. Report me the problem

Q: I've got an AGA computer and SWOS patch only runs the ECS version. Why?

A: Because only ECS version is supported at the moment. Hang on.

Q: When I run a formerly frozen game, using snapshot/RESUME option (registered ←
    only),
    the game does not work properly, or the sound is not correct.

A: Remove the RESUME tooltype, start the game normally, exit, then set RESUME ←
    again
    and retry.

Q: Why can't you promote the games to multiscan or GFX cards. You can promote them ←
    to PAL/NTSC
    already.

A: This would suppose that I include cybergfx code in JST, because most of games ←
   run with the
   OS trashed, and the games use directly hardware registers and copperlists. ←
      Sorry that's
   impossible, unless you run the patches under UAE :)

Q: Couldn't you add modem/ethernet support to null-modem games (like Overdrive) ?

A: That's be great but I'd have to code a full serial.device/ethernet device ←
   handling in JST,
   and then patch the games to support this protocol. You'd better get a long ←
      cable :)

Q: Why some of your patches do not exit using F10 or HELP, or any other key?

A: Because JST can be used as a degrader to run old DOS compatible games, and most ←
    of those
   are so ashamed they trashed the memory bad that they don't want to quit, so JST ←
       can't do
   anything about this.

Q: Why some of your patches don't iconify/snapshot?

A: the old DOS compatible games fall into this category (for the reason see answer ←
    just above),
   some pure JST loaders don't bear it, and for the others, I've got to add some ←
      code to
   each of them, because this operation needs copperlist pointer snooping, which I ←
       can achieve
   only by acting slightly but specificly on every loader.

Q: I tried to install your patch of <game> but the install tells me something like ←
    "decode disk
   error" or "sync not found". Why?

A: Either you're trying to install a cracked version of a non trackdos game, or ←
    you're trying
   a trackdos rerelease (or coverdisk) of a non trackdos game. If you've got a ←
      crack, no need
   to write to me...

Q: I've got an unsupported version of Pinball Dreams

A: Check my WWW page. A freely downloadable version of the game is there.

Q: JST does not work well with my new PPC board

A: Impossible, since Ralf would have noticed it!

Q: Lemmings locks sometimes after 5 or 6 levels

A: Not anymore. Get the latest version on aminet or my www page. There were 9 code
   checksum which made the program behave weird when tampered with.

Q: I cannot install a single game using rob2file (Team17 games and some others)

A: Try rob2file V2.0. It's completely system friendly and should work with all ↩
    drives,
   even with the drives from AT A1200, which was not the case with rob2file 1.x.
   Games which fall in this category are:

- Chaos Engine AGA (Team17)
- Xenon 2 (Kixx/Bitmap Bros)
- AlienBreed 2 AGA (Team17)
- ArcadePool (Team17)
- Assassin Special Edition (Team17)
- BodyBlows ECS (Team17)
- BodyBlows Galactic ECS/AGA (Team17)
- Apache (Team17)
- Overdrive (Team17)
- Project-X Special Edition (Team17)
- Superfrog (Team17)
- Apache (Team17)
- Mortal Kombat (Akklaim)
- Mortal Kombat2 (Akklaim)
- Out To Lunch AGA (Mindscape)

Q: Why using a JST loader and OSEmu to run a game while it runs OK
   with AmigaDOS?

A: Because it allows to configure the memory you need, suppresses
   problems due to commodities, and allows to quit back to WB.

Q: The game runs but crashes or gfx is corrupt, or there are slowdowns

A: Try setting MMU=NONE or/and NOCACHES

Q: When I run a game from the CD, or from HD, sometimes JST locks while
   loading the files (it's not a game with diskfiles, but with real files)
   The computer does not crash, but sometimes the WB freezes too (no refresh)
   and the partition where the game is is not available till I reset.

A: I'm aware of the problem, and I noticed it since I upgraded to Kick 3.1,
   but I don't know if there's a connection. Even JST 1.3 sometimes locks, so
   it's not an "improvement" I added recently! I completely rewrote the
   directory scanning/file loading system to try to eradicate the problem,
   and it still occurs from time to time. I don't know what I'm doing wrong.

Q: Why was the WHDload emulation coded? To play a nasty trick to Bert?
   What's the use of it?

A: No! I had to code some basic functionnalities at first when I began OSEmu
   integration in JST (OSEmu only knows WHDload calls). Then, I thought that
   it would be interesting to code WHDLoad emulation because the work to do
   was almost done.
   This emulation seems to be very useful, because some people have problems
   with WHDLoad (with JST too!!), and can only run the slaves using JST
   emulation, because they don't know how to use WHDLoad and are too lazy, or
   they've got a configuration that WHDLoad does not like.
   If the game fails with JST and WHDLoad, then they're sure there's
   another problem. Some people are used to JST and don't want to learn WHDLoad,
   or appreciate the many features of JST. But to be honest, I appreciate the
   stability of WHDLoad compared to JST nasty crashes :)
   I also found some bugs in JST when I wondered why some WHDLoad slaves did
   not run in the emulation, so it can be very useful for all JST loaders.
   Think about it.

Q: Is JST certified for year 2000?

A: Bite me

Q: What does FAQ does mean?

A: Frequently Asked Questions.

1.10   JST Degrader

You can use JST with an executable that has not been designed to work with it, and ←
    with which
you've got trouble because of the display or the caches, VBR, etc...

The degrader mode of JST is activated by the EXECUTE tooltype.

If you specify EXECUTE, JST will try to execute the object you passed, degrading ←
    the
display to PAL (or NTSC if you select NTSC), zeroing the VBR (unless you specify ←
    LEAVEVBR),
disabling 68060 specific caches (unless you set LEAVECACHES) and setting the ←
    sprites to OCS size
(required by most games). Of course, loader-specific options like LOWMEM, HDLOAD, ←
    NOFAST,
SAVEDIR and DELAY will not work.

If you want to disable the caches, add the NOCACHES argument.

You can also set JST as a default tool in the executable icon (set icontype to ←
    Project)
just like WBKillAga.

1.11   CD32 joypad emulation

Since V2.9, JST is able to emulate CD32 joypad with joystick and keyboard.
A separate tool called Fakepad is also downloadable from my page.

To use this emulation, you must set FAKEPAD tooltype to 1,2 or 3, and the EXECUTE
tooltype (JST must be in degrader mode).

The games using lowlevel.library are completely fooled by this trick.

Keymaps:

FAKEPAD=1:

Blue: F1 key
Green: F2 key
Yellow: F3 key
Play: F4 key
Left ear: F5 key
Right ear: F6 key

FAKEPAD=2:

Blue: F7 key
Green: F8 key
Yellow: F9 key
Play: F10 key
Left ear: HELP key
Right ear: '*' key (num keypad)

FAKEPAD=3

Blue: 'E'
Green: 'R'
Yellow: 'T'
Play: 'Y'
Left ear: 'U'
Right ear: 'I'

FAKEPAD activated (value 1, 2 or 3):

Directions: joystick
Fire (red): joystick fire button
Blue: joystick 2nd button (if any)

Example of CD32 games using lowlevel.library (and to which this option may be ←
    applied):

- D-Generation
- Robocod
- Super Street Fighter II Turbo
- Gunship 2000
- Worms

1.12   WHDLoad compatibility

From v1.3, JST now supports the WHDLoad slave format. That means that you can ←
   theorically play
any game patched using the WHDLoad method with JST instead by simply replacing the ←
    WHDLoad
default tool by JST in the icon.

WHDLoad compatibility is best for V6 or V7, but JST can run some V10 slaves.

Of course, compatibility is not 100% (and it's not intended to be 100%, get ←
   WHDLoad
itself !)
Here's a list of the games which work using JST as WHDLoad emulator:

If you cannot make a game run,try with NOVBRMOVE. It works sometimes.

Due to some unsupported WHDload commands in JST, a game may need to have
the NOCACHES tooltype set, whereas it works without it if you use WHDLoad.

There are some problems when the games use OS swap for gamesaves load/save, but
I'm working on this.

Some are quite okay:

9 Fingers (no quit)
World Of Commodore (no quit)
Hardwired (no quit)
Vision Megademo IV (no quit)
Arte (no quit)

Some do not work at all (some crash badly):

The black list

1.13   HRTMon compatibility

Bert Jahn and Alain Malek worked on compatibility between WHDLoad and HRTMon,
for instance for the freezer key, or special WHDLoad commands callable from HRTMon ←
    .

The good news is that owing to WHDLoad emulation in JST, JST inherits the
HRTMon WHDLoad features:

- Setting FREEZEKEY tooltype allows to enter HRTMon if installed
- HRTMon "wq" command also quits a JST loader

But:

- MMU commands ProtectReadWrite, etc... still not supported.

Some people still say that WHDLoad emulation is useless in JST... :)

Notes:

– Minimum HRTMon version required: v2.22
– NOVBRMOVE tooltype prevents HRTMon activation

1.14 OSEmu - AmigaOS emulation

OSEmu is an AmigaOS emulation module originally written by Harry and
maintained/improved by Mr Larmer, Bert Jahn and me.

The advantages of OSEmu are multiple:

– you can run a game without booting without startup-sequence, making
  some shitty startup scripts to set assigns, disable commodities...
– the game is independent of the version of the ROM on your computer
– you can quit to Workbench anytime by simply pressing the numerical '*',
  or if you're a registered user, you can set QUITKEY to the
  required value. In the original game, you could not quit to WB!
– OSEmu can run many CD32 games. Calls to cd.device, cdtv.device are
  ignored and do not crash or cause errors (the game thinks the CD is there,
  but of course you don't hear anything!) and lowlevel.library is partially
  supported, which means that you deceive the game by faking a joypad in
  game ports. Gamesaves using nonvolatile.library are not emulated yet.

In order to use OSEmu with JST, the file OSEmu.400 must be located in
the same directory as the JST executable.
Some game install packages using OSEmu will propose you to update your
version. At the moment, there's no automatic version check in the installers,
but you can check OSEmu versions by using the "version" command.

First, OSEmu was written to run A500 games, but now it can run CD32 games,
and can emulate the joypad with function keys:

F1: blue
F2: green
F3: yellow
F4: play/pause
F5: left ear
F6: right ear

Of course joystick fire button is equal to red CD32 joypad button, and if you've
got a 2 differentiated button joystick, the second button matches the blue CD32
joypad button.

OSEmu is free and the source can be found either on the WHDLoad site
or in the JST developper package.

Check OSEmu version by typing "version C:OSEmu.400" (or whereever OSEmu is ←
    installed)

1.15 Disclaimer

I'm not responsible for any damage caused directly or indirectly by this piece
of software.

The shareware version of JST and my HD work can be distributed freely.
However, I'd appreciate
that you drop me a line if you include it in a coverdisk or PD collection.
If you like (or you dislike) this piece of software, you can e-mail me.

The registered version of JST should NOT be distributed.

1.16   Author

Regular mail:

Jean-François FABRE
19 Rue Emile Duployé
34500 BEZIERS
FRANCE

E-mail:

jffabre@free.fr

PLEASE don't send big files (>500Ko) to my internet address unless I allow you ←
    to.
It's got the property to act on my nerves in a negative way. Thanks.

WWW:

Visit my patch page, where all my HD work is gathered, along with some other
quality installers (all for original software), and some other interesting
related freeware stuff:

http://jffabre.free.fr/amiga/patches.html

FTP:

Available in all good aminet sites retailers (game/patch) and grocery stores.

FAX:

Yes, thanks. I'd like one.

Cellular phone:

They'll *NEVER* make me buy one of those!

1.17   Developpement team

It's not really a team, but rather the people who wrote code that was
used in JST.

- Me (main developper)
- Ralf Huvendiek (co-developper, added MMU features, fixed loads of bugs)
- Kenneth Nielsen for the startup code
- Joannes Geiss for the hunk decoding code
- Alain Malek for CIAs and keyboard reset code.

1.18   History

                    V3.2b
        - Fixed (definitely!) tooltype bug parsing (from WB)
        - Fixed (I think) random lock bug and start from CLI bug (at least!!!)
        - Fixed WHDLoad ListFiles emulation. Now OSEmu & WHDLoad emu will work ←
           better
        - If a WHDLoad slave is loaded, SAVEDIR is set to current directory by ←
           default
          (SAVEDIR environment variable overridden, just as with OSEmu)

V3.2
        - Fixed this CLI argument bug again. arrgh
        - Fix for 68030 MMUCode users. Please reinstall MMUCode

V3.1
        - Bug removed in degrading part, present since v3.0beta (locks at startup)
        - MMU tooltype is now compatible with HRTMon
        - HRTMon is called (if active) if a MMU hit is encountered
        - DEADLY tooltype removed. Now this is default. Please report loaders which
          don't work anymore (if any)
        - MMU tooltype changed. Now takes 3 values: NONE,LAXIST and FULL (default)
        - LEAVEMMU tooltype removed (now equates MMU=NONE)
        - Fixed behavior when called from CLI (more stable now I think)
        - Default copperlist location moved from 0 to another chip address
          (because some poorly programmed programs expected a 0 in 0!!)
        - 0 address forced with 0
        - Bugfix in custom chips restoration
        - Added some nice functions for developpers (but I'm about the only one
          and anyway I did not document the new features :( )
        - Improved WHDLoad compatibility. Now Indy 500 works.
        - RESUME feature still does not work :(

V3.0
        - Fixed ReadArgs() buffer problem
        - Added support for hunk debug

V3.0 beta (beta but worth upgrading!)
        - Registered users only: If OSEmu is active, the SAVEDIR environment
          variable is ignored (set to "")
        - LEAVEVBR option only available with EXECUTE option
        - Bugfix: The snapshot save was broken
        - Bugfix: after a restart after iconify, it was not possible to enter ←
           HRTMon
        - Bugfix: in degrading function (VBR)

```
            - Bugfix: stack management (caused nasty hanghups)
            - Added OSEmuReloc() function (developpers contact me for autodocs upgrade)
            - Added HexToNum() function
            - Harmless (?) enforcer hit removed in argument parsing
            - Still to do for V3.0:
                * keyboard acknowledgement
                * RESUME function broken. Disabled in this release
```

V2.9d
```
            - Better timer restore (thanks Luca for the track!)
            - Better keyboard acknowledgement.
            - This will be the last version before next year :)
```

V2.9c
```
            - Big bugfixes in WHDLoad emulation
```

V2.9b
```
            - Small code changes (bsr -> jsr). That's all but I had to identify
              a new version while awaiting for v3.0.
```

V2.9
```
            - FAKEPAD tooltype/argument added for CD32 joypad emulation in games
              runnable by EXECUTE option (degrader mode)
```

V2.8b
```
            - Better WHDload emulation (now Jurassic Park works)
```

V2.8
```
            - Added RelocateExecutable function. Added Relocate WHDLoad support
              but still no luck with Total Carnage and Moonstone...
            - Some misc things fixed
```

V2.7e
```
            - Directory scanning improved. Now not limited to one directory level.
              But still locks sometimes, and I really don't understand why.
            - Fixed keyboard acknowledgement on exit
```

V2.7c
```
            - Misc improvements (I don't remember, actually :))
```

V2.7b
```
            - Improved file readind/loading
            - Added chipmem clear on startup (thanks Harry)
```

V2.7
```
            - Fixed freezes with MMUCode on 68040 (removed MMU detection, so if you've
              got a EC040, don't try the MMU tooltype !)
            - Improved file loading. Still locks in some cases.
            - Consumes less memory on non MMU systems (removed MMU boundary alignment
              if MMU is not active)
            - New tooltype FREEZERMB added. Allows to enter HRTMon with the right mouse
              button (sometimes the keyboard is just dead!)
            - JST now first looks up the file "OSEmu.400" in the game directory before
              looking it up in JST PROGDIR:
```

V2.6b
```
            - Aaagh, problems with loaders allowing iconify/snapshot since V2.4
```

```
            probably due to a phxass bug. Fixed. Thanks to James Battle for signaling
            this bug! (ex: Supercars 2, Supercars...)
```

V2.6
```
        - Expansion memory support added for loaders using OSEmu
          (needs v1.4 of OSEmu, included in this archive)
        - OSEmu version displayed on loader bootup
```
V2.5:
```
        - Added WHDLoad expansion memory support (WHDLoad v8)
        - Fixed WHDLoad mode bug
        - Fixed some buggy VERBOSE traces
        - Now EnterDebugger() works before SaveOSData() (crashed before)
```

V2.4b:
```
        - GetFileLength() did not work with LOADDIR (Sensible Soccer CD32)
        - Stupid empty lines display removed in message console
          (buggy since V2.4)
```

V2.4:
```
        - Aaaagh! WriteFilePart() was completely buggy! Fixed.
        - Some misc bugs fixed
        - Barfly gave me lots of trouble (assembly bugs). I'm sticking
          with PhxAss (not crashing anymore since I splitted the source files)
        - WriteFilePart() improved: all files are updated in the cache
        - More WHDLoad games work (Batman The Movie, Great Giana Sisters)
```

V2.3b:
```
        - WriteFilePart() improved: diskfiles are updated in the cache
          when a writing occurs: useful for OSEmu (see Arcade Pool)
        - NOOSSWAP option prevents OSEmu from writing to the disk
        - Code modifications because I'll soon compile using Barfly
          (bug crash/guru problems with PhxAss v4.39, too bad!)
```

V2.3:
```
        - Nasty bug fixed in the unregistered version. Thanx Stuart.
        - Small bugfixes
        - The GetFileLength() function was not available
        - New function GetUserFlags() added (developpers)
```

V2.2:
```
        - Size limit for WriteFileHD and WriteUserFileHD set to 4MB
        - Bug fixed in MMU mode (affected some loaders)
        - Bug fixed on WriteFilePartHD. SWOS CD32 properly save games now.
        - New developper function DisableChipmemGap() added
```

V2.1b:
```
        - Correct MMUCode again
        - InstallWizard v1.2 included
```

V2.1:
```
        - Autodocs added, besides the functions.guide file (developpers pack)
        - Install-script fixed (user package)
        - New tool Install-Wizard added (developpers pack)
          No documentation yet, but it's rather self-explanatory
        - A few bugs fixed by Ralf
        - MMU code changed by Ralf
```

```
V2.0:
        - Ralf implemented MMU handling for all CPUs which own one:  ←
           68030,68040,68060.
        - Ralf debugged iconify/snapshot which was broken
        - Ralf improved a lot of other things. JST works a lot better now.
        - And me? I must admit I did not do a lot on JST, but registering people  ←
           and
          writing loaders.

V1.5:
        - Merged both registered and shareware versions. Now a personal keyfile
          allows to use the registered features with the unique JST version
          (easier JST updates, distribution, ...)

V1.4b:
        - Fixed a bug which prevented file writes to be done properly
          (the bug appeared in v1.4)
        - HRTMon detection changed. Even if novbrmove is set in HRTMonPrefs,
          HRTMon is detected.
        - FREEZEKEY and SAVEDIR are now configurable by environment variables
          (registered users only)

V1.4:
        - Ralf:
           reworked source code a lot.
           added a freezerkey (FREEZEKEY tooltype) and a EnterDebugger function ( ←
              for HRTMon)
           added full MMU support for 68030 (68040/68060 support still to come)
           added ReadFileFromImage function (to read a file from a OFS diskimage)
           improved SaveOSData by handling USP too
           cleaned up chip memory test
           fixed PowerPacker decruncher
        - Jeff:
           improved LoadFiles so subdirs are also scanned and read
           fixed HRTMon detection by JST (compatible with HRTMon v2.22 and higher)
           (see
                HRTMon compatibility
                 for further details)
           fixed a bug in LoadFiles (sometimes the routine locked the OS)
           partially implemented WHDLoad Decrunch routine. Aladdin slave now works ←
              OK
           repaired broken OBJNAME tooltype
           added full french documentation

V1.3a:
        - bugfix for WHDLoad loaders

V1.3: (major release)
        - fixed a bug which prevented some loaders to run (ReadFile return code)
        - added Harry's OS emulation support (UseHarryOSEmu)
        - added WHDLoad slave support (!!!!)
          (Beta feature, see the
                  compatibility list
                  to know the games which work)
        - added QUITKEY tooltype
        - added NOVBRMOVE tooltype
        - added many developper functions (CRC16, ReadFilePart...)
```

        – allows LOWMEM with AGA games

V1.2a:
        – fixed a bug preventing the use of DEADLY

V1.2 (beta2):
        – Better LoadFiles() function. No more locks with FFS 44.4 and faster!
        – Debugged LOADDIR (once again!!)
        – Ralf reworked the source code, and I merged both versions.
        – Better exception handling, autoquit key features (developpers).
          (I think this one still needs
          some testing, but I could not wait more to release it)

V1.1f:
        – fixed broken DEADLY option. Thanks to Jacob Jensen for reporting it!

V1.1e:
        – fixed some bugs in LOWMEM/HDLOAD and LOADDIR combined together
        – fixed some bugs in diskfile/file error handling (no locks, error & quit ←
          instead)
        – Ralf fixed some potential bugs and corrected the OS swap routine (less ←
          freezes)
        – Ralf relocated the VBR, autoquit key is due for soon (programmers ←
          interest)

V1.1d:
        – LOADDIR bug fixed (registered users only)
        – Potential code relocation fixed (thanks to Ralf)
        – Enabled wrong virtual disk id again (thanks to Ralf)

V1.1cR2:
        – rob2file completely rewritten system-friendly style. Many people had ←
          problems
          with some Team17/Probe games that did not install (** Wrong key or no ←
            disk in drive...)
        – Package reworked. Install script for JST and disk tools.
        – Note: the JST executable was not changed (which explains the 1.1c release ←
          2)

V1.1c:
        – Now MMU translations are disabled on 68040 and 68060 to allow JST to run
          with new 68060.library versions and ROMs mapped in Fastmem. If you ←
            experience
          problems on exit (guru), try with LEAVEMMU on. No MMU support for 68030 ←
            yet, sorry.
        – Added HexToString in developper functions.

V1.1b:
        – Keypresses during LOWMEM no longer hang JST (don't tease it, tough :))
        – Added 2 very convienient functions for programmers:
          possibility to log the patches done and write it to a file and
          automatic blitter search/replace/blitwait insertion for move.w #addr,($58 ←
            ,Ax)

V1.1a:
        – Improved snapshots (some things to add with some loaders)
        – Reload command from iconify menu added (registered only)

```
V1.1:
      - RESUME tooltype (registered version) allows snapshots on disk! (for  ←
        loaders
      which support iconify option)
      - Menu in iconify mode to allow to save snapshot, quit, resume
      - Keyboard acknowledged properly on iconify (no more annoying chars)
      - Small OS swap improvements
      - Modified BeamDelay routine. Contact me in case of timing problems, music  ←
        not
      playing right...

V1.0:
      - Added iconify feature (the loaders must be designed for it, because of  ←
        the
      read-only copperlist register!)
      - Updated developper documentation to support iconifying.

V0.9c:
      - Ooops fixed a bug in argument parsing from CLI (the first time it happens ←
        ?)
      - Added LOADDIR tooltype (registered users only)
      - Added some programmer functions like TestDirectory.
      - A few loaders will need to be slightly rewritten (get the Street Racer  ←
        update
      and Space Hulk update).

V0.9b:
      - Added NOQUIT tooltype (to use with EXECUTE)
      - Bug removed: in case of abnormal termination, JST did not execute the  ←
        post
      script (registered version only)
      - Added LOADDIR tooltype, but function not implemented yet (registered  ←
        version)

V0.9a:
      - Oooops, fixed a bug: if T: did not exist, the scripts were executed and  ←
        your
      HD could be unvalidated because of the T creation (by calling "execute")
      (only registered users who used V0.9 were "exposed")
      - TestDirectory function added (developpers)

V0.9:
      - added the 2 scripts jstpre.sh and jstpost.sh (registered version only)
      Very useful to disable commodities/ethernet cards/enforcer before JST  ←
        executes
      the program/loader and to activate them again when JST exits.
      - for developpers: problem when the nbdisks argument is accidentaly set to  ←
        0
      in the HD_PARAMS macro

V0.8f1:
      - Aaaaaaargh argument another parsing problem (not with registered version)
      prevented the usage of most of the options either from WB or CLI

V0.8f:
      - Arghh, some games like Lure Of The Tempress don't like
```

           some gfx fixes during OS swaps. Fixed.

V0.8e:
        – Oooops, HDLOAD did not work anymore. Fixed now.
        – Default interrupt 2 (keyboard) improved
        – OS swap routines improved
        – Accurate MOVEP exception detection on 68060

V0.8d:
        – Bug/limitation that prevented some loaders to work removed: SetFilesPath ←
          added.

V0.8c:
        – Ooops. NTSC switch enabled again
        – Thanks to Keith, now some display problems on Pyramax, Puzznic,
          and some others fixed. So degrader mode is also more efficient now.

V0.8b:
        – USERDATA tooltype added for loaders which need it
          (useful for rippers) and also to pass args to a command
          if you use EXECUTE to degrade an executable
        – GetUserData function added to read this data
        – StrncpyAsm function added (D2 sets max string length)
        – Now blitter mask is cleared by SaveOSData, which simulates better
          a cold boot.

V0.8a:
        – Fixed bug with HDLOAD and a subdirectory as data dir

V0.8:
        – LEAVECACHES tooltype added: no caches will be disabled, not even
          68060 specific ones.
        – EXECUTE tooltype added to allow to execute normal DOS programs
          (like KillAGA) with 68060 support. Other tooltypes can be used
          with it too (LEAVEVBR, LEAVECACHES, NOCACHES, NTSC).
          Now JST can be used as a degrader
        – Sprites are set to lores using the system at startup (thanx to Bert
          Jahn for the routine).

V0.7:
        – Simple execbase emulation added (developpers)
        – Functions added to handle this emulation (developpers)

V0.6b (internal release):
        – Bug removed: In HDLOAD mode JST could not load file in another directory
          than current one. This is now fixed (only Kick 2.0+, sorry)
        – Nasty bug found: if not enough mem for quit the game crashed because
          of wrong stack relocation. Some people may have noticed the problem.
        – Bug removed in interrupt aknowledge
        – Removed long delay at startup in case of DELAY on.
        – FIRE decruncher support added
        – PatchExceptions behaviour change. Now can be used again in program to ←
           trap
          exceptions only (programmers only)

V0.6a:
        – Changed default stack location.

```
V0.6:
        - Removed BENICE tooltype
        - Removed joybutton press = LOWMEM (for people using joymouse)
        - Added DEADLY tooltype
        - Changed behaviour in case of unexpected interrupts. Now tries to  ←
          acknowledge
          the interrupts unless DEADLY is on. In this case, exits with an error
          (thanks to Carlo Pirri for the accurate bug reports)
        - Bugfix in 'registers.dat' file creation. Now extension memory is properly ←
          reported
        - Added another stop sound command on quit
        - assembled with optimize option

V0.5c:
        - Bugfix in ReadFileHD for getlength command
        - ReadUserDir() function added
        - Now battery backed up clock is reloaded only on a quit from the game

V0.5b:
        - bugfix in SaveOSData: ExecBase was not cleared if VBR was 0 or LEAVEVBR  ←
          was on
        - bugfix in arguments parsing on shareware version. Almost all tooltypes  ←
          did not work!
        - stack relocated to top of used chipmem in SaveOSData: Agony, BodyBlows  ←
          original
          & BodyBlows Galactic work on a standard A1200/020/NoFast, but some other  ←
          loaders
          needed to be recoded to work (Prince, Lotus...)
        - all registers D0-A6 set to 0 in SaveOSData
        - sprite dma disabled at bootup to avoid some display corruption

V0.5a:
        - DMA set/clr bugfix in os swap routine
        - WaitBOF function added (not functionnal yet)
        - Better WaitBlit() routine

V0.5:
        - Bugfix: intuition screen was not opened!
        - More accurate exception handling (useful for coders/bug reports)
        - Added TPWM decrunch routine (experimental, not tested)
        - Changed ImploderDecrunch(). Now only registers D2-D5/A2-A4 are saved.

V0.4b:
        - Sprites/display are reset better when an OS swap occurs (HDLOAD/LOWMEM  ←
          mode)

V0.4a:
        - JST now opens an intuition screen of 4 colors (vs 16 in older versions)
        - Now JST does not allow system calls after SaveOSData (programmers only)
        - BENICE tooltype added for compatibilty (users who have problems with old  ←
          loaders)
        - Removed a bug in argument parsing from CLI

V0.4:
        - Now JST loads the clock from battery backed up clock if found (registered ←
          only)
```

```
V0.3 : JST is now SHAREWARE. Some features are only accessible to registered users ←
  .

        - 1.3 compatibility restored. SAVEDIR still unsupported
        - VERBOSE tooltype added.
        - RNC\02 unpack method added to RNCDecrunch()
        - Alloc24BitMem() function added (for some games which do not like 32 bit  ←
          fastmem)
        - Read/Write/DeleteUserFileHD() functions added to support SAVEDIR
        (developper(s?) please update your loaders which load/save user data to  ←
          disk)
        - Slightly changed DMA handling on os swaps
        - Fixed a bug in dump register functionnality (DEBUG option)
        - Now the display comes up and the dumps are written after (DEBUG option).
        - SAVEDIR tooltype added (registered only)

V0.2e:
        - internal release

V0.2d:
        - NOOSSWAP no longer forbids HDLOAD or LOWMEM. It's now a user flag

V0.2c:
        - Fixed bug in LoadDisksFromName (Cadaver)
        - Changed task name to "JOTDStartup"

V0.2b:
        - Changed intuition screen depth from 8 colors to 16

V0.2a:
        - fixed bug with LoadDisksIndex and LOWMEM (Superfrog intro)

V0.2:
        - first aminet release, few details changed

V0.1:
        - developper release
```

1.19 Acknowledgements

```
I'd like to thank the following people:

- Keith Krellwitz (Abaddon) for interesting talk about loaders and for writing  ←
  about 1/3
  of the loaders using JST !

- Alain Malek (Hornet) for HRTMon and for his source code

- Ralf Huvendiek for having choosed JST to code some loaders, and for helping me  ←
  to improve
  JST. He added lots of cool features and found a couple of nasty bugs already.
```

– Harry for having initiated the OSEmu project (originally to make Another World ←
   run)

– Johannes R. Geiss for the original hunk relocation code (which was a bit buggy ←
   :))

– Dave Jones for his Hunk guide.

– All the registered users in general, and in particular Robert Plutchak, Chris ←
   Vella, Steven Velletri,
   Wolfgang Unger.

– Ninjaw P.B. for having written the Amiga Patches Guide

– Harry, John Girvin, Bert Jahn, Codetapper, Mr Larmer and all the WHDLoad coders ←
   for their loaders and interesting chat.

– People who report bugs ACCURATELY :)

– Your name here: _____