

JOTD HD Install Collection Documentation

COLLABORATORS

	<i>TITLE :</i> JOTD HD Install Collection Documentation		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 31, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	JOTD HD Install Collection Documentation	1
1.1	JOTD Startup Documentation	1
1.2	Introduction	1
1.3	Getting Original Games	2
1.4	Registering JST	2
1.5	Arguments/Tooltypes	3
1.6	OBJNAME	4
1.7	NTSC	4
1.8	TRAINER	4
1.9	QUIET	5
1.10	NOCACHES	5
1.11	LEAVECACHES	5
1.12	NOFAST	5
1.13	HDLOAD	5
1.14	RESUME	5
1.15	EXECUTE	6
1.16	FILTEROFF	6
1.17	FAKEPAD	6
1.18	USERDATA	6
1.19	LOWMEM	6
1.20	NOOSSWAP	7
1.21	NOQUIT	7
1.22	DELAY	7
1.23	SAVEDIR	7
1.24	LOADDIR	8
1.25	QUITKEY	8
1.26	ICONIFYKEY	8
1.27	DEADLY	8
1.28	NOVBRMOVE	8
1.29	Debug and Development	9

1.30 MMU Settings	10
1.31 Start and End Scripts	10
1.32 Snapshot Handling	10
1.33 TroubleShooting	11
1.34 FAQ	12
1.35 JST Degrader	15
1.36 CD32 Joypad Emulation	15
1.37 WHDLoad Compatibility	16
1.38 HRTMon Compatibility	17
1.39 OSEmu - AmigaOS Emulation	18
1.40 Disclaimer	18
1.41 Author	19
1.42 Development Team	19
1.43 History	20
1.44 Older Version History	24
1.45 Still to do	31
1.46 Acknowledgements	32
1.47 Tools	32
1.48 Disk2File	33
1.49 Rob2File	33

Chapter 1

JOTD HD Install Collection Documentation

1.1 JOTD Startup Documentation

JOTD Startup V4.7b
© Copyright 1995-2001 Jean-François Fabre/Ralf Huvendiek

Introduction	Tools
Arguments/Tooltypes	Register
CD32 Joypad Emulation	Getting Original Games
JST Degradar	Disclaimer
WHDLoad Compatibility	Acknowledgements
HRTMon Compatibility	FAQ
OSEmu - OS Emulation	Author
Start and End Scripts	Development Team
Snapshots	History
TroubleShooting	Future

1.2 Introduction

As you have probably noticed, 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 developed 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 features.

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).

The latest versions of JST have almost perfect WHDLoad emulation so there are literally hundreds of games that can now be installed using JST.

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.

From v4.7, as JST is now free, the RESUME feature is available to everyone.

JOTD HD Startup is © 1995-2001 Jean-François FABRE

JOTDStartup is © 2001 Jean-François FABRE

1.3 Getting Original 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 these days, and that's why I offer my services to help you:

- If your original disks are damaged. Please send me the original disks which are damaged (as proof of purchase) and I'll will try to rewrite to them or replace them with new disks. Please send money to cover the costs of duplication and packaging (and more if you like :). Please contact me before sending your originals, as I may not have the disks myself 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.
- If 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 the box and manual. Contact me for any request about that.

1.4 Registering JST

JOTDStartup is FREeware!

From v4.7, due to many factors, I decided to release JST for free, with full source code. Here are the main reasons:

- Some undiscovered bugs still show up (lockup while loading files...)
 - Lots of trouble with PPC/Appolo boards and/or some mysterious patches
 - Smart bastards saying that JST always crashes. They've got the source. They can ← debug.
-

- Nice people wanting to help in JST debugging (and maybe extensions)
- No f****in' pirate succeeded in generating a fully working keyfile for JST :)
- If I'm gobbled by a giant chicken, someone else can continue my work
- I was ashamed by the poor support I was able to give to registered users who had ↵ trouble
- I made enough money with that piece of shit (about 150 registrations in 4 years)
- WHDLoad is more stable, WHDLoad works better, blah, blah ...
- Ok, shut the f**k up, JST is 20DM cheaper now :)
- You *HAVE* to see RESUME working without paying :)

Registered Features now available to everyone

- Snapshot reload. RESUME
- Possibility to save and load games in a different directory than the game directory. SAVEDIR and LOADDIR
- Quiet mode. QUIET
- Reload battery backed-up clock on exit.
- Configurable iconify key. ICONIFYKEY
- Configurable quit key QUITKEY when the loader supports it.

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 icon tooltypes.

OPTIONAL tooltypes will only work with loaders that support them.

GENERAL tooltypes have an effect on all the loaders.

DEGRADING ONLY tooltypes are used when degrading.

Here are all the tooltypes in full:

OBJNAME	FAKEPAD
NTSC	USERDATA
TRAINER	LOWMEM
QUIET	NOOSSWAP
NOCACHES	NOQUIT
LEAVECACHES	DELAY
NOFAST	SAVEDIR
HDLOAD	LOADDIR
RESUME	QUITKEY
EXECUTE	NOVBRMOVE
FILTEROFF	ICONIFYKEY

DEBUG/DEVELOPMENT TOOLTYPES

MMU SETTINGS

Some options can be set by an environment variable:

Those are FREEZEKEY and SAVEDIR.

e.g. setenv FREEZEKEY \$5C, setenv SAVEDIR RAD:

EXAMPLES from CLI for normal JST objects:

```
JST MagicPocketsHD NOCACHES DELAY NTSC
```

The game doesn't like caches and the disk is a little slow. The screen has also been promoted to NTSC.

```
JST HookHD NOFAST HDLOAD SAVEDIR=Ram:
```

The game has had fastmem turned off and the save games will be saved to the ram disk. Memory is saved by reading directly from the disk.

EXAMPLES from CLI for normal DOS executables:

```
JST Dune NOCACHES EXECUTE
```

The game does not like caches.

```
JST ssboot EXECUTE USERDATA=SWOS
```

The ssboot needs degrading, but also needs swos to be passed as an argument.

```
JST AGADemo LEAVEVBR LEAVECACHES NTSC EXECUTE
```

The demo likes caches and does not care about the VBR being 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, the display features of JST will be cancelled. JST's no David Copperfield.

1.6 OBJNAME

OBJNAME (OPTIONAL)

Allows you to specify an object name (icon launch).

1.7 NTSC

NTSC (GENERAL)

Changes the screenmode from the default PAL to NTSC.

1.8 TRAINER

TRAINER (OPTIONAL)

Activates the trainer, if there is one.

The trainer will probably give infinite amounts of lives, time, etc...

1.9 QUIET

QUIET (GENERAL)

Turns off all the output of JST. No window or error messages are shown, so only use after you have tested that the loader works correctly.

1.10 NOCACHES

NOCACHES/NOCACHE (GENERAL)

Disables the CPU caches, by setting CACR to 0. Supports the 68060.

1.11 LEAVECACHES

LEAVECACHES (GENERAL)

Leaves the cache settings alone. Disabled by NOCACHES .

1.12 NOFAST

NOFAST (GENERAL)

Disables the fast memory and so always allocates extension memory from chip memory. If you get crap sprites, this could help, but it's useless in the case of 512K mem games.

1.13 HDLOAD

HDLOAD (GENERAL)

Saves some memory by loading the data directly from the HD instead of caching all the files in RAM. It works for diskfiles and normal files, but it's useless in games with only 1 diskfile, use LOWMEM instead.

It can cause the screen to "black out" while the computer is loading from the HD.

1.14 RESUME

RESUME (OPTIONAL)

Resumes 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.

1.15 EXECUTE

EXECUTE (GENERAL) (DEGRADING ONLY)

Allows the use of JST as a normal degrader like KillAGA, but with 68060 support and more control over the system.

1.16 FILTEROFF

FILTEROFF (GENERAL)

This tooltype will merely force the filter off at startup.
This can be useful if your WB is configured with filter on and you want games (like Jaguar XJ220) to run without the filter.

1.17 FAKEPAD

FAKEPAD (DEGRADING ONLY)

Runs the CD32 joypad emulation so that owners of joysticks or just a keyboard can play games that need a CD32 joypad.
FAKEPAD=1 will use the 1st keyset, FAKEPAD=2 will use the second, and so on. There are 3 different keysets.

1.18 USERDATA

USERDATA (DEGRADING ONLY)

If a string is passed to this argument and the EXECUTE tooltype is selected, then this string will hold the arguments of the degraded program.

e.g. JST ssboot EXECUTE USERDATA=swos

The ssboot is degraded, but it needs "swos" as an argument.

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.

Is also used in some JST loaders to specify parameters.

1.19 LOWMEM

LOWMEM (OPTIONAL)

Prevents the caching of diskfiles in memory. JST will 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 an error message or crash with an exception. In that case they don't support this option yet :)

1.20 NOOSSWAP

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 and 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, your hard drive can become invalidated.

1.21 NOQUIT

NOQUIT (DEGRADING ONLY)

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).

1.22 DELAY

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).

1.23 SAVEDIR

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. You should also use LOADDIR .

1.24 LOADDIR

LOADDIR (OPTIONAL)

Allows the user to specify a directory where the game files or diskfiles should be loaded from. You should also use `SAVEDIR` .

1.25 QUITKEY

QUITKEY (OPTIONAL)

The raw keycode of the wished exit key should be passed to this tootype. If the loader supports it, this key is now the quit key.

e.g. `QUITKEY=$5F` sets `HELP` as the quit key.

NOTE: Does not work if `NOVBRMOVE` is activated.

1.26 ICONIFYKEY

ICONIFYKEY (OPTIONAL)

The raw keycode of the wished iconify key should be passed to this tootype. If the loader supports it (or if `SNOOPCUSTOM` is on, this key is now the iconify key.)

e.g. `ICONIFY=$58` sets `F9` as the new iconify key.

NOTE: Does not work if `NOVBRMOVE` is activated.

1.27 DEADLY

DEADLY (GENERAL)

Most 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 `DEADLY` off (and so keeping `execbase` alive) can be unreliable and can trigger 81000005 GURUs if 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.

1.28 NOVBRMOVE

NOVBRMOVE (GENERAL) (68010+ ONLY)

Prevents the use of 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).

This causes incompatibilities with ICONIFYKEY
 QUITKEY
 HRTMon

1.29 Debug and Development

DEBUG (GENERAL) : Only useful for bug reports. Makes a memory image on exit.

LEAVEVBR (GENERAL) : Only useful 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.

SNOOPCUSTOM (GENERAL) : This tooltype allows custom registers snooping. Used with DEBUG, it writes the file "custommemory.dat" where the values of all custom registers modified during the game are stored. Unmodified registers will have a value of zero. The custommemory.dat file is binary but easy to read. For instance, at offset \$80 you'll find the copperlist pointer, and at offset \$9A you'll find the interrupt enable register.

This tooltype can be VERY useful in conjunction with the ICONIFYKEY one. It will make iconify possible even if the author of the loader had not implemented iconification in his slave (e.g. all WHDLoad loaders!!!!!!!)

Warning: this can slowdown the games a lot.

1.30 MMU Settings

The MMU can be used in one of 3 ways:

- 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. This is the default.

1.31 Start and End Scripts

You can make JST run scripts on startup and when you quit the loader. Useful to disable/enable commodities, ethernet cards, enforcer, etc...

The 2 scripts are run every time JST loads/exits. They should be located in S: and be called:

```
jstpre.sh - pre-JST script
jstpost.sh - post-JST script
```

They should contain commands for turning off and on problem software that could cause JST problems.

NOTE: The virtual disk T: must exist for this feature to work.

1.32 Snapshot Handling

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...

1.33 Troubleshooting

The latest versions of JST all have built-in troubleshooting help. If there is a problem, JST will try to tell you what you should do.

- 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 the `EXECUTE` tooltype/argument.
- Not enough memory: try to free some memory. Playing with `NOFAST` and `HDLOAD` and booting 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 executed. 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. Mail 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. 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/loaders on the following Amiga set-ups:

A500/68000/2Megs	(small game like POP)
A1200/Blizzard III 68030/18Megs	(the one I've just bought)
A1200/Blizzard 68060/64Megs	(the one I'm developing with)
A1200/68040/34Megs	(the one from a friend)

Some users reported problems with Apollo cards and some low level programs. I'm not able to handle this ATM. You'll have to boot without startup-sequence and without setpatch to avoid trouble!!

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

1.34 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 RESUME option 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 would 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 be better getting a long cable :)

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

A: JST can be used as a degrader to run old DOS compatible games, but most of those are poorly programmed and they trash the memory badly so that the games just can't quit. JST can't do anything about this.

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

A: The old DOS compatible games fall into this category (for the reason see answer just above). Some pure JST loaders can't handle it, and for the others, I've got to add some code to each of them.

The operation needs copperlist pointer snooping, which I can only achieve by acting slightly but specifically on every loader.

However, from v4.0 of JST by setting the SNOOPCUSTOM tooltype along with the ICONIFYKEY tooltype, you can iconify/snapshot more games, and even WHDLoad emulated slaves!!!!. But the SNOOPCUSTOM option slows down the game when it does not make it crash (IK+). This is interesting anyway.

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.

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.

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 and/or 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 fonctionnalities 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: The MMU=NONE feature appears to be broken on my 68060 and never works correctly when setpatch is installed. Why?

A: There may be a problem with parts of the system remapped to fastmem, or your 060.library may not be 100% compatible. I might add MMU.library support which would probably fix this bug.

Q: Is JST certified for year 2000?

A: Bite me.

Q: What does FAQ does mean?

A: Frequently Asked Questions.

Q: i found a Cyberguard hit in JST vX.X

Here it comes:

BYTE-READ from 00000000

PC: 09BBD650

USP: 089147B8 SR: 0004 FLSW: 01210200 TCB: 08903FC0 Data: 00000000

00000000 00000000 00000000 00000000 00000000 0223414D 08BFC844 Addr:

00000000 09BBF460 08BFDA8A 08BFC844 09BC6000 00F959E6 080008BA 089147B8

It doesn't crash, but hangs the miggy leaving me only to reboot.

BTW: I CAN play the game, (it was FLASHBACK) but only if i DON'T start CyberGuard. Well, you don't need that prog, you say. That's right, so i believe that error isn't very heavy!

A: JST is not compatible with Cyberguard, enforcer or any other memory enforcement program. The system part of JST is (or tries to be) free from hits, but as soon as JST tries to take the system over, hits happen. I don't want to try to make JST Cyberguard compatible because it's rather useless (who uses CyberGuard/Enforcer 100% of the time on the Amiga?)

1.35 JST Degradar

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, zeroing the VBR, disabling 68060 specific caches and setting the sprites to OCS size (required by most games). Tooltypes such as NTSC, NOVBRMOVE and LEAVECACHES will cancel the effect of the above features..

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.

1.36 CD32 Joypad Emulation

Since V2.9, JST has been able to emulate CD32 joypad with a joystick and the keyboard. A stand-alone version called Fakepad is also downloadable from my website.

To use this emulation, you must set FAKEPAD tooltype and the EXECUTE

tooltype (JST must be in degrader mode). Games using lowlevel.library are completely fooled by this trick.

There are 3 different keyboard configurations:

FAKEPAD=1

Blue : F1 key
Green : F2 key
Yellow : F3 key
Red : F4 key
Left Ear : F5 key
Right Ear : F6 key

FAKEPAD=2

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

FAKEPAD=3

Blue : 'E'
Green : 'R'
Yellow : 'T'
Red : 'Y'
Left Ear : 'U'
Right Ear : 'I'

ALL CONFIGURATIONS

Directions : Joystick
Fire (Red) : Joystick Fire Button
Blue : Joystick 2nd Button (If any)

Examples of CD32 games using lowlevel.library (and so work with Fakepad):

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

1.37 WHDLoad Compatibility

From v1.3, JST supports the WHDLoad slave format. That means that you can theoretically 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 most of V11 slaves. The compatibility is constantly improving :)

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 the `NOVBRMOVE` tooltype. 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

1.38 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.
 - See the settings I'm using with HRTMon 2.2x
 - In the settings, you'll see IDE harddrive is disabled. Enabling it will cause access faults in WHDLoad and you'll have to set `MMU=NONE` or `LAXIST` to make it work with JST
-

1.39 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 using 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 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:

Blue	: F5 key
Green	: F6 key
Yellow	: F7 key
Red	: F8 key
Left Ear	: F9 key
Right Ear	: F10 key

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 developer package.

Check OSEmu version by typing "version C:OSEmu.400".
(Or wherever OSEmu is installed).

1.40 Disclaimer

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

JST binary, source and my HD work can be distributed freely. However, I keep intellectual property for all that stuff.
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.

1.41 Author

Regular mail:

Jean-François FABRE
19 Rue Emile Duployé
34500 BEZIERS
FRANCE

E-mail:

jffabre@free.fr

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'lll *NEVER* make me buy one of those!

1.42 Development Team

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

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

1.43 History

V4.7b

- Fixed an old bug in file reading (return code was not correct)
Now my WHDLoad converted slaves work, which was not the case previously.

V4.7

- JST is now FREE. Full source code is provided in developer package
- Awesome WHDLoad now works

V4.6b

- NOSSWAP enforces no os read/write again (useful for OSEmu)
- Fixed potential crash on startup for users who have a SSP remap utility
- Fixed display which could cut the last pixels sometimes

V4.6

- Fixed potential serious bug in low-level string copy routine (StrcpyAsm() ←)
- Fixed program end code (Permit() removed)
- InGameExit() can be called even if the system is up
- Added GetLoadDir() function
- Removed "Insert Volume ..." system requesters

V4.5c

- Fixed crash at the end of a program run with EXECUTE
- Optimized CopyMem a bit more
- Better 68060 exception handling
- Better 68000 support

V4.5b

- Oooops, sorry, In v4.5, I've broken something that P5 060 libs did not like at all in order to make RESUME work with MMU. Now fixed.
- Changed of assembler for the 54536th time because Barfly crashed.
Now JST is built using phxass v4.39 (and the exe is smaller too)
- Fixed version check problem present since very early versions of JST
- Supports WHDLoad slaves v13

V4.5

- RESUME now works with MMU (no more need for MMU=NONE) and reload option now works too! I'm happy.
- Improved VBR handling. Should be more compatible now
- VBR improvement now fixes the ShapeShifter/JST conflict (thanks Luca!)
- JST is no longer Kick 1.x compatible. I think this is not necessary ← anymore
now, and saves a lot of hassle.

V4.4

- Reduced stack usage (SWIV works at least!, and possibly others)
- Added CBSWITCH_SET WHDLoad flag for better compatibility (also for OSEmu)
- Package contains debugged OSEmu (the write to disk function did not work)
- Fixed debug output (registers.dat)
- Added traps for addresses \$100-\$400 (just in case some trouble with PPC boards would be related to that)
- Fixed some write cache update stuff (for gamesaves)

V4.3b

- Oops, fixed big memory leak problem (100k!) if started from Workbench (bug appeared in v4.2c)
- Added more debug information

V4.3

- Added keymap WHDLoad tag (needed for some WHDLoad slaves)

V4.3 beta

- Switched from PhxAss to Barfly asm
- Code cleanup
- Misc changes

V4.2c

- Changed startup code
- Other misc changes. Should be a little more stable now
- Highly enhanced OSEmu.

V4.2b

- Repaired OSEmu initialization (some diskimage based games could not run)

V4.2

- Added WHDLoad emulation DeleteFile() (OSEmu needs it)
- Repaired broken DeleteFileHD
- ReadDir now also returns directories (OSEmu needs it)

V4.1

- Fixed WHDLoad emulation+OSEmu read dir bug
- Added potential support for decrunched files in~some games
- Removed MMU valid marking for IDE interface (remove IDE option in HRTMon ← !)
- HDLOAD was broken. Now fixed.
- ReadDir() function was buggy. Fixed. Now OSEmu Examine()/ExNext() work
- Misc bugs fixed
- Greatly improved OSEmu version.
- 2 serious Enforcer hits removed

V4.0

- RESUME is back! but only works with MMU=NONE at the moment (Phase 5 libs can cause problems with MMU=NONE)
- Tooltype FILTEROFF added
- Tooltype FORCECLIST added. Experimental.
- VERBOSE was broken in 4.0 beta. Fixed
- MMU tooltype was broken in 4.0 beta (forced to FULL). Fixed
- WHDLoad tag CBAF added
- WHDLoad support for some slaves in v12 format
- WHDLoad Patch routine implemented (Cannon Fodder 2 new slave)

V4.0 beta (still beta because RESUME still disabled)

- Major code cleanup
 - No loader worked on 68020/NoFast since a long time. Now fixed.
 - DEBUG tooltype was not tested from WB. Fixed
 - Added WHDLoad Delay function (PipeDream)
 - Degrading code cleaned up
 - Better stack handling
 - Better cache flush
 - Better file cache handling (mostly for OSEmu loaders)
 - Write-thru cache implemented (dynamic file cache entry creation/resize)
-

- if file is created or written during an OS swap). Only available if MMU=NONE or MMU=LAXIST at the moment because of a technical problem
- If OSEmu loaded, user & supervisor stack are allocated and thus protected
- SAVEDIR environment variable override now forced to LOADDIR and no longer to "" for WHDLoad slaves and OSEmu loaders (more logical)
- DEBUG tooltype was not checked. Fixed
- Fixed error message: cannot file disk ...
- Fixed unwanted exits on Cannon Fodder 2 slave (WHDLoad emulation)
- Improved CopyMem() function: now tries to work with longwords: means: faster AND now Robocop WHDLoad works (strange chipmem byte write & DMA conflict problem)!!
- Bugfix in directory/path handling: now Cannon Fodder 2 WHDLoad works 100%
- Improved register dump
- Smaller executable
- Guidefile reworked again, with many links added (thanks Kristian)

V3.4c/d

- Degraded mess up. Bugfixes. Sorry for that!!

V3.4b

- Aaaargh, typing error, problems on exit: JST tried to write a dump file! Now fixed. It's the only change from v3.4

V3.4

- Major code cleanup
- Various memory leaks in MMUCode fixed
- Led state now properly restored on exit (thanks Luca for noticing & testing it!)
- Fixed access fault that could occur during writes to diskfiles (Arcade Pool)
- Misc (but sometimes important) bugfixes
- All icons converted to OS3.5

V3.3 (beta, unreleased as non-beta)

- Fixed some blitter initialization stuff: Lemmings intro now displays OK
- Removed some interrupt stuff which wasted CPU time for nothing -> less speed problems on games (Chaos Engine AGA, Space Hulk...)
- Better WHDLoad emulation
- Fixed nasty bug for EXECUTE option (locked the system on exit)
- Better OSEmu included in the package. Many functions added
- Fixed nasty bug in WriteFileHD (Space Hulk crashed JST because of that)
- Fixed AttnFlags for 68040/68060: removed the AFF_68881 and AFF_68882 as FPU emulation does not work when the OS is down.
- Fixed version strings (thanks Luca!)
- Changed stack default location

V3.2c

- Fixed Access Fault when saving files
 - Better MMU protection (for ReadFileFast, i.e. games with normal files)
 - Start and end scripts were not executed anymore (thanks Christian)
 - OSEmu was not loaded when in the game directory. Fixed.
-

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)
- Apparently a lockup which sometimes occurred with some ethernet cards has gone too (damn this version IS bugfixed :))

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 behaviour 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 developers (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 hangups)
- Added OSEmuReloc() function (developers 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

Previous Version History

1.44 Older Version History

Newer Version History

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 reading/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:

- Aaaaagh! 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 developer 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 developer functions.

V1.1b:

- Keypresses during LOWMEM no longer hang JST (don't tease it, tough :))
- Added 2 very convenient 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 developer documentation to support iconifying.

V0.9c:

- Oops 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 (developers)

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 developers: problem when the nbdisk argument is accidentally 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: SetFilePath ↔ 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 acknowledge
- 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 & 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 functional 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 compatibility (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
(developer(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 functionality (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:

- developer release

1.45 Still to do

I try to work on JST as soon as I've got some spare time, but there are still some things to do on it:

- Bring back the RESUME feature (I'm close, but still no luck after hours of work).
 - Fix remaining incompatibilities with WHDLoad loaders.
 - Add a better acknowledge keyboard routine on exit (WHDLoad's got the same problem).
 - Add mmu.library support.
 - Reduce memory consumption.
 - Add a default quit key for WHDLoad loaders.
-

1.46 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 all the loaders using JST! Too bad he stuck to the PC and his work. But he sent me all his loaders and source code, so I can continue fixing bugs in his loaders.
- Alain Malek (Hornet) for HRTMon and for his source code.
- Ralf Huvendiek for having choosen 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. Too bad he reduced the pace, but one has to work :)
- Harry for having initiated the OSEmu project (originally to make Another World run), Mr Larmer and Bert Jahn for having improved it widely.
- Harry (again!) for interesting technical and friendly talks, and for the effort of semi-automatic JST source -> WHDLoad conversion
- Johannes R. Geiss for the original hunk relocation code (which was a bit buggy :))
- Dave Jones for his Hunk guide.
- Dr Zarkov for his JST installs and adaptations.
- Bored Seal for his installs and bug reports.
- Kristian Hesketh for greatly improving the guide file you're currently reading.
- All the former registered users in general, and in particular Robert Plutchak, Chris Vella, Steven Velletri, Luca Longone, and 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.
- Christian Sauer for his tests and bug reports with ethernet card and more...
- People who report bugs ACCURATELY :)
- Your name here: _____

1.47 Tools

There are a number of tools which are supplied in the JST archive for installing all the available games.

Disk2File : Standard DOS track reader.
Boot2File : Standard DOS boot reader (tracks 0-1).
Rob2File : Rob Northen 12 sector disk imager.
Psydir : Psygnosis format 1 directory maker.
RipPsyFiles : Psygnosis format 1 files ripper.
Grem2File : Gremlins format files ripper.
Raw2File : Raw track imager.

Tools in bold are only part of the developer package.

They are all now system friendly and should work with any disk drive attached to your Amiga. DF0-DF3.

1.48 Disk2File

Disk2File is a disk imager for normal (almost) copiable DOS disks.

It's used in many installs scripts using JST. Now it will not be included with new HD loader packages anymore. Put it in your C: directory.

Usage:

```
Disk2File <unit> <file> <starttrack> <endtrack> <SKIPROB>
```

<unit> : Take the last digit from DFx (e.g. DF0) and use it here.
<file> : Name and path of the diskfile to create.
<starttrack> : From 0 to <endtrack>.
<endtrack> : From <starttrack> to 159.
<SKIPROB> : Flag to tell Disk2File to skip track 0 head 1
(but tracksize bytes will be written).

Disk2File is Kick 1.x/2.x/3.x compatible.

1.49 Rob2File

Rob2File is a disk imager for Rob Northen protected disk format.

It's used in many installs scripts using JST. Now it will not be included with new HD loader packages anymore. Put it in your C: directory.

Usage:

```
Rob2File <unit> <file> <starttrack> <endtrack> <diskkey>
```

<unit> : Take the last digit from DFx (e.g. DF0) and use it here.
<file> : Name and path of diskfile to create.
<starttrack> : From 0 to <endtrack>.
<endtrack> : From <starttrack> to 159.
<diskkey> : Hexadecimal key. Depends on the game. Default is 0.

Rob2File is Kick1.x compatible.

Since V2.0, rob2file is completely system friendly.

List of games using rob2file to install:

- Chaos Engine AGA (Team17)
 - Xenon2 Kixx 1 disked (Team17)
 - 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 Kombat 2 (Akklaim)
 - Out To Lunch AGA (Mindscape)
 - Primal Rage (but the loader does not exist!)
-