
DAAD Adventure Writer

Version 2 - Release 3

A multi-machine adventure writing system.

Revised in March 2019

Project lead, coding, English language recovery and testing,
debugging, bundled tools, documentation:

Tim Gilberts, Stefan Vogt

Syntax highlighter:

Chris Ainsley

Spanish language testing, worked example (modern adaption),
countless contributions:

Pedro Fernández

A special thanks goes to **Andrés Samudio** who kindly contributed DAAD to the
public domain, allowing us to build upon this wonderful heritage.

<http://8-bit.info/infinite-imaginations-aventuras-ad/>

Preamble

Growing up in Europe in the 80's with an interest in adventure games, whether you just wanted to play or write, meant directly or indirectly using the Gilsoft adventure systems. With "The Quill" and its successor "PAWs (Professional Adventure Writer)", Gilsoft published groundbreaking applications that made it possible to create machine code adventures in a simple and innovative way. For most of us it was only the finished games that we experienced, but some of us took advantage of the new possibilities and wrote interactive novels. A very well-known early advocate of the systems was, for example, Fergus McNeill (Bored of the Rings, The Colour of Magic, The Big Sleaze).

While PAWs was the last system open to the public, it was not the last system involving Tim Gilberts, the founder of Gilsoft. After the Gilsoft label faded in the late 1980s, he founded Infinite Imaginations to advise and support other companies with customized tools and solutions. During this time, SWAN (System Without A Name) was created for Fergus McNeill's Delta 4 and it was the first system only available to a single company.

Then came DAAD. It is the final system based on the code that can be traced all the way back to Quill. At the same time, it is the most advanced system in this chain, supporting a parser that is easily on par with Infocom titles. DAAD was developed as an in-house solution for the legendary Spanish adventure forge Aventuras AD. After Aventuras AD had to close on bankruptcy in 1992, the system was considered lost for many years. In 2014, Mr. Samudio, the founder of Aventuras AD, discovered the system in his attic. Instead of keeping DAAD under lock and key, he made it available to the Spanish adventure scene. His generous public domain contribution is the foundation on which this new release of DAAD is based.

When the DAAD was available again after decades, it turned out that time was not good with the system. Theoretically, adventures were supported in both Spanish and English. Practically it was so that the English interpreters were erased on almost every medium. Since they were not used by Aventuras AD, they had been removed over time. The same applied to the English game templates. It seemed that English language support was gone missing in the perpetual tides of time. In addition, the C64 disk was corrupted. Parts of it could be restored, but the Spanish parser was no longer the last version but a rip from the last Aventuras AD game Chichén Itzá. One could arrange with the system, but it was a fact that major parts of DAAD would probably remain lost forever.

As part of their ongoing collaboration in preserving the Gilsoft heritage, Tim Gilberts, creator of the DAAD Adventure Writer, and Stefan Vogt decided to fully recover the system and to finally make it available to the English language audience. Instead of only restoring the missing files, they decided to also make useful additions to the system, with the modern retro adventure author in mind. The fruits of this collaboration is what you have downloaded now.

What is DAAD and what can it be for you?

DAAD is a multi-machine and multi-graphics adventure writer, allowing you to target a broad range of 8-bit and 16-bit systems, including C64, ZX Spectrum, Amstrad CPC, MSX, PCW, Atari ST, Amiga, IBM PC (DOS). You may create as many adventure games with the system as you want to, even commercial ones. Before you start using DAAD, we suggest however to carefully read the legal section at the end of this document.

DAAD is very sophisticated but please don't expect it to be the PAWs 2 that never was in store. DAAD was created with the professional adventure developer in mind. Rather than being a single application, it is a set of more than 30 tools that need to be operated from the DOS command shell. DAAD requires programming skills and knowledge in handling source code files and compilers. The sources of your adventures (.SCE files) compile to game databases which need to be mastered and transferred to the target systems where they run in platform-specific interpreters. The tools necessary to achieve this are provided or referenced. There is a section in the 1991 documentation called "a worked example" which explains the necessary steps. In addition, we have a worked example section in this 2018 documentation, with the purpose to complement the original issue with modern knowledge and easier ways of mastering your adventure game.

It is highly recommended that you have experience with the Gilsoft adventure systems, especially with the CondActs logic of Quill and PAWs, because a superset of this is the foundation of DAAD. Perfect would be knowledge about the PAWs CP/M version. It is the Gilsoft system which is the closest to DAAD. Note that the .SCE files from CP/M PAWs are similar to the .SCE files DAAD uses, but they are not the same. However, you may consider our project ANTUR, a PAWs to DAAD transcompiler, allowing you to port an existing CP/M PAWs adventure to DAAD. ANTUR is available here: <https://github.com/ByteProject/antur>.

System requirements and recommendations

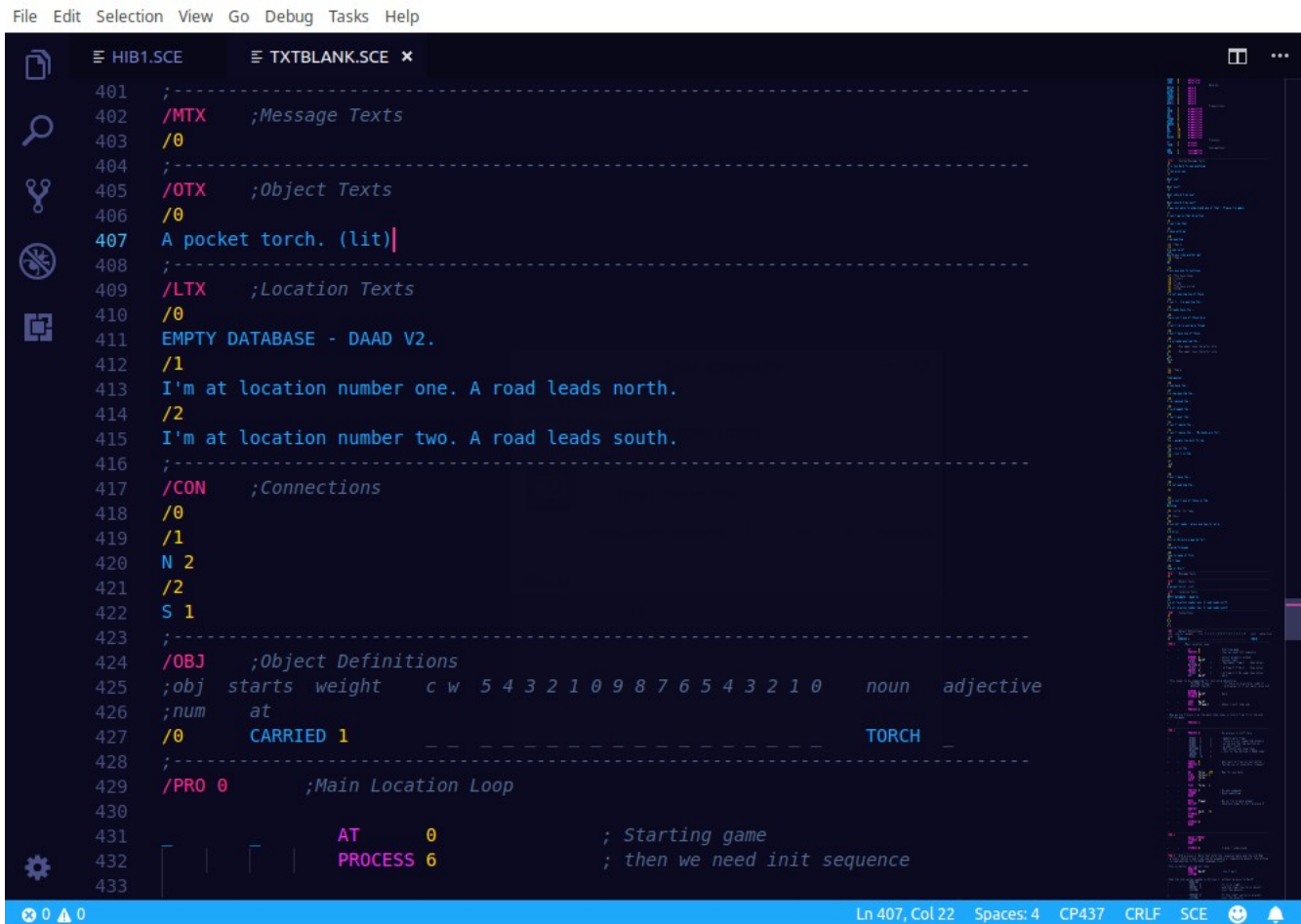
We wanted to ensure that you can work with DAAD in a (mostly) modern environment. Here are the mandatory system requirements to start developing your own adventures:

- a modern operating system (Linux, MacOS, Windows)
- Visual Studio Code
- DOSBox

To properly setup Visual Studio Code, you should also install the .SCE Syntax Highlighter that Chris Ainsley created for this project. You may download the extension from the VSCode Marketplace: [SCE Syntax Highlighter \(DAAD/PAWs\)](#).

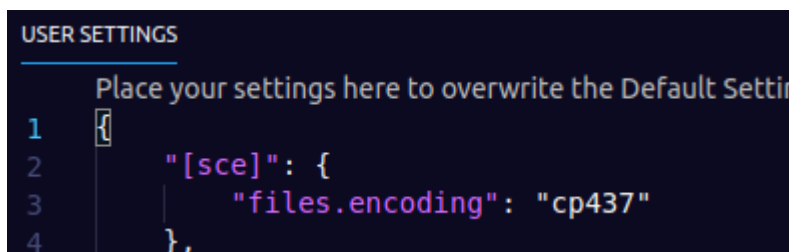
We highly recommend using it together with the gorgeous “Outrun” theme, that resembles the wonderful colours of the 80s. It conveys the proper retro look and feel. Get Outrun from here: [Outrun theme](#).

This is how a SCE file will look after you applied these changes:



```
File Edit Selection View Go Debug Tasks Help
HIB1.SCE  TXTBLANK.SCE x
401 ;-----
402 /MTX ;Message Texts
403 /0
404 ;-----
405 /OTX ;Object Texts
406 /0
407 A pocket torch. (lit)|
408 ;-----
409 /LTX ;Location Texts
410 /0
411 EMPTY DATABASE - DAAD V2.
412 /1
413 I'm at location number one. A road leads north.
414 /2
415 I'm at location number two. A road leads south.
416 ;-----
417 /CON ;Connections
418 /0
419 /1
420 N 2
421 /2
422 S 1
423 ;-----
424 /OBJ ;Object Definitions
425 ;obj starts weight c w 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 noun adjective
426 ;num at
427 /0 CARRIED 1 TORCH
428 ;-----
429 /PRO 0 ;Main Location Loop
430
431 AT 0 ; Starting game
432 PROCESS 6 ; then we need init sequence
433
```

Note that DAAD .SCE files need to be processed in DOS. To ensure compatibility, we suggest you to open them with the code page 437 encoding. You can override the default settings in user settings to always open .SCE files with CP437 encoding.



```
USER SETTINGS
Place your settings here to overwrite the Default Settings
1 {
2   "[sce]": {
3     "files.encoding": "cp437"
4   },
5 }
```

After you mounted the contents of the DAAD directory to DOSBox you're fully setup and ready to create your own interactive novels.

Changes and additions in DAAD v2 Release 3 (2019)

in **DAAD**

- DC.EXE → new DAAD compiler version 2.42, which significantly improves text compression and reduces DB sizes
- LEGACY.EXE → the original DAAD compiler 2.40 from 1991
- BACKUP → creates a backup from a specified .SCE file on drive Y: (we advise to connect a folder from your Dropbox to that drive letter in DOSBox)

in **LIB\ST**

- DAAD_Atari.ST → new English language interpreters (EDI) that fix both a major and a minor bug

in **LIB\C64**

- DAAD_Loader.d64 → Commodore 64 (5.25 Disk) template for DAAD adventures with loading screen

in **LIB\CPC**

- DAAD_Loader.dsk → Amstrad CPC (3" Disk) template for DAAD adventures with loading screen

in **LIB\SPECTRUM**

- DAAD_SLoader.dsk → Spectrum +3 (3" Disk) template for DAAD adventures with loading screen

in **DAAD\TAPMAST**

- png2scr.py → creates Spectrum loading screens from PNG images
- dump-pal.py → creates CPC palette files from PNG images
- DLPART1.BAS → CPC tape loader template for DAAD adventures with loading screen, loads PART1
- DLPART2.BAS → CPC tape loader template for DAAD adventures with loading screen, loads PART2
- DAAD.FNT → the DAAD standard CPC font (this file is used by the CPC tape loader)
- SPECCY.BAS → Spectrum tape loader template for DAAD adventures with loading screen

in **DAAD\TOOLS**

- MARS → merges an Amstrad CPC mode 0 image and a palette file to DAAD's CPC loading screen format

Changes and additions in DAAD v2 Release 2 (2018)

in **DAAD\SCE**

- BLANK.SCE → recreated English language database template
- BLANK.DDB → compiled English language database template
- TXTBLANK.SCE → English language database template for text-only games
- SPANISH.SCE → Spanish language database template
- SPANISH.DDB → compiled Spanish language database template
- HIB1.SCE → the complete source of Pond's Hibernated 1 (read file!)

in **DAAD**

- MOVEDB → moves PARTx.DDB database files to TEST directory
- RUN → quickly runs a game via DOS interpreter(s) for testing

in **DAAD\TAPMAST**

- TAPCAT → creates ZX Spectrum tapes (.TAP files) for distribution
- 2CDT → creates CPC tapes (.CDT files) for distribution

in **DAAD\TOOLS**

- ACHTUNG → adds a Commodore 64 header to a database
- SC → allows editing C64 disk image files (D64)

on the **DAAD C64 disk** in **LIB\C64**

- LE1 → loader for an ENGLISH game PART1
- LE2 → loader for an ENGLISH game PART2
- [c]NEWCHRSET → alt. charset taken from Chichén Itzá by Aventuras AD
- EDI → new version of the ENGLISH C64 interpreter
- SDI → new version of the SPANISH C64 interpreter

on the **DAAD CPC disk** in **LIB\CPC**

- DCPCIE.Z80 → recovered ENGLISH CPC interpreter

on the **DAAD compile disk** in **LIB\CPC**

- GFX.BIN → minimal CPC graphics database for text-only games

The DAAD compile disk is an image which quickly allows you to create your final adventure binaries for distribution. We also added a **CP/M Plus** image, as the MCRF tool which merges the runnable files, is CP/M based. Details in the new worked example section.

in **LIB\AMIGA**

- PART1.DAT → minimal Amiga graphics database for text-only games
- S-PIC.ADF → Amiga tool to create startup screens from IFF images

on the **DAAD Spectrum disk** in **LIB\SPECTRUM**

- PART1.SDG → minimal Speccy graphics database for text-only games
- BLANK.DDB → compiled Spectrum database template for testing
- DS48IE.P3F → recovered ENGLISH Spectrum interpreter
- TMASTER.BAS → loader script for Aventuras AD tape master tool
- TBOOT2.BIN → Aventuras AD tape master tool (recovered, untested)
- DRE.BAS → loader script for ENGLISH +3 game
- MERGE → basic script, merges all files into single binary ENGLISH
- MERGES → basic script, merges all files into single binary SPANISH
- TAPLOAD → template for a simple tape loader with SCREEN\$

in **LIB\ST**

- PART1.DAT → minimal Atari ST graphics database for text-only games

Directory structure DAAD v2 R3 (2019)

We significantly changed the directory structure and the bundled files compared to the incomplete release in Spain a few years ago. The reason behind this was the intention to provide a ready to use distribution. We also wanted to separate the actual system files from the historical important heritage from Aventuras AD. That's why there are two separate DAAD downloads available from the official website at: <http://8-bit.info/infinite-imaginations-aventuras-ad/>

→ The DAAD Adventure Writer

Contains all the tools to develop adventure games.

→ DAAD Aventuras AD preservation files

Contains historical files from Aventuras AD that were rediscovered together with DAAD, mostly Spanish game master disks and sources.

- DAAD** → DAAD root dir where the compiler (DC.EXE) files are located. It's recommended to put the contents of this directory into the root of a DOS drive, D:\ for example
- DAAD\TOOLS** → contains all the system utilities in one place, you should add this folder to your PATH variable in DOSBox.
- DAAD\TEST** → quick DOS test environment for your adventures
- DAAD\TAPEMAST** → directory for mastering tape files (Spectrum, CPC)
- DAAD\SCE** → contains all the database template files and examples
- DAAD\OBJ** → as referenced in the 1991 documentation, no changes
- DAAD\INTERP** → contains all the DOS interpreter files, see 1991 docs
- LIB** → interpreters and tools to roll out your games on the supported platforms
- DOCS** → the documentation

Image editing – pixel graphics



The DAAD 1991 documentation often refers to the well-known ST program DEGAS for editing pixel graphics and loading screens. While it's fine to still use DEGAS, we suggest you to take a look at some of the modern and convenient solutions. Here are two programs we highly recommend, **GrafX2** as replacement for editing PI1 files (the common ST format), **Multipaint** for editing loading screens for Spectrum, C64, CPC and MSX. It is very easy to bundle a loading screen with the 16-bit versions of your DAAD games (Commodore Amiga, Atari ST and DOS), which you will notice that when you come to the worked example sections. Adding a loading screen to the 8-bit versions is much harder to achieve so we created DAAD loader templates for all 8-bit machines, which also extend DAAD's functionality. The process how to use these 8-bit templates is described in a separate chapter after the worked example section of this documentation.



A worked example in modern times

Please note that the DAAD compiler **DC.EXE** has been updated for the first time since 1990. The new version is 2.42. While the old compiler did only compress location texts, the new compiler also supports compressing messages and system messages, which has a significant impact on the database sizes and thus allows you to create bigger games.

The [-c] switch of the DAAD compiler now has options:

- 0: do not compress
- 1: compress locations
- 2: compress locations and messages
- 3: compress locations, messages and system messages

Example: **dc MYFILE.SCE PART1 -c3 -l0 -m0** would compress LTX + MTX + STX

Note: **-l0** is the English language compiler option. DAAD's default language is Spanish, so make sure to always add language 0 for compiling an English game.

The old compiler version 2.40 remains as **LECACY.EXE** in the DAAD root directory.

ATARI ST

This is one is quite easy. Do exactly as the 1991 documentation says. Don't do the suggested cable transfer though. Use an emulator of choice to get your game files on an empty disk image. **Hatari** is a good solution as it allows mounting directories as TOS hard drives. Don't forget to copy the interpreter files from the DAAD ST disk image: EDI for English and SDI for Spanish adventures.

AMIGA

Compile as described in the 1991 documentation. Don't use any of the transfer programs unless you really want to do it the oldschool way. We highly recommend getting yourself **ADF Opus**, a great explorer and editor for Amiga disk images. It is a Windows application but works fine under Linux with Wine. Make a copy of the MinOS Amiga disk image (in LIB\AMIGA\). Replace PART1.DDB on the image with your own game database. Do the same with the PART1.DAT file if your adventure is not txt-only. If you don't want to add a loading screen you're basically done already. Go ahead like this if you want to add one: instead of creating a loading screen as described in the documentation, create an IFF image with a modern tool like **GIMP**. Use the provided S-Pic utility (see bundled documentation) to create a compressed executable from your image. Add it to your game disk and add the image executable name as an entry in "s/startup-sequence". It needs to be entered before the line that's loading the interpreter. The English interpreter (EDI) had been renamed to INTERP in the Minimum OS template. If you're creating a Spanish game, you need to delete the interpreter that's already on disk and copy SDI from the DAAD Amiga disk to your game disk. Rename it to INTERP and you're done. Copying and renaming can all be done with **ADF Opus**.

PC (DOS)

No changes to the 1991 documentation. We just want to add that you definitely should stick with the method described as “using new system multi-machine graphics”. After 28 years it’s safe to say that this method will work best for you.

CPC

You can stay close to the 1991 documentation. We recommend using **CPCDiskXP** to transfer your game files to an Amstrad disk. CPCDiskXP is a Windows application but it will run perfectly with Wine under Linux. On Linux we recommend also **Arnold** as the primary Emulator, for Windows there are quite a few emulators available, so pick the one you like the most. In LIB\CPC is a minimal disk that is setup with the essential files ready to compile your game (DAAD_compile.DSK). You should always make a copy of the compile-disk first rather than using the original image. This has the advantage that you can delete all the files except your game binary from it after the compilation process completed. The disk contains a file GFX.BIN which is an empty graphics database you can use for text-only adventures. Replace it with your own file if you created graphics. DAAD always wants graphic files, it won’t work without. Compilation is handled by MCRF which is a CP/M program. So you need to boot into CP/M for the compilation itself. The 1991 documentation says that you may use CP/M 2.2 and CP/M 3 (Plus), which is wrong unfortunately. Only CP/M 3 (Plus) will work. You’ll find an image in LIB\CPC. Here is the synopsis:

MCRF outfile{.BIN} interp{.Z80} text{.DDB|.BIN} graphics{.BIN}

Note that you must specify the type of text database. DDB is from the compiler direct (which is recommended) and BIN is assumed to have a CPC disc header.

After you created a native CPC binary, you may use the Windows-Console application 2CDT.EXE (in DAAD\TAPMAST) to also create a tape file. It will work with Wine CMD under Linux. Careful though, as 2CDT.EXE is not a DOS executable. The docs for 2CDT are in the directory. To grab your game file from the Amstrad disk image use CPCDiskXP, or alternatively the console utility iDSK.

ZX Spectrum

This workflow is a bit different to what it was in the past so we completely replace the 1991 documentation with this.

Compile a version of your source without debug information compressed called xxx.DDB (etc) using option -m1. Use ASH to add a Spectrum header to the .DDB file, rename it PARTx.DDB. Make a copy of your DAAD_Spectrum.dsk. Use **WinAPE** to transfer the game files to it. Don’t be confused we are using an **CPCDiskXP** again to move files to a Spectrum +3 disk image. Both machines use the same disk system so it will work. Save and close CPCDiskXP after transferring the files. Now open the image in your Spectrum emulator. We recommend **ZEsarUX** or **Fuse**. Make sure you selected a +3 machine for emulation.

You are already prepared. We recommend to go with the brand-new DAAD loader template, usage is explained in the next chapter of the documentation. The old loader templates on the DAAD Spectrum disk are still there but deprecated.

You probably want to distribute a TAP file as these can be easily used to create real tapes and may also be used in common interfaces like the divMMC. The steps to achieve this are somewhat different from a +3 release. On the DAAD Spectrum disk image, you find a file called MERGE (and MERGES) for Spanish games. Run it via LOAD "MERGE". It will load the interpreter, PART1.DDB and PART1.SDG into memory. The Basic prompt will reappear after that happened. Now use the following command to save the memory contents to disk as a single executable:

Now you need a tape loader. We recommend to use the advanced DAAD tape loader template. It's properly described in the next chapter of this documentation.

The Commodore 64 was our problem child due to corrupted disks, outdated interpreters, missing headers. We are going to replace the 1991 docs completely with process: Compile a version of your source without debug information compressed called PART1.DDB (etc) using option -m2. Use our shiny new tool ACHTUNG (only Tim can tell you what the acronym stands for and yes, it is one) to add a C64 header to your database. The output format of ACHTUNG is *.DDC. Rename the file to PART1 without any suffix. Make a copy of the DAAD_C64.D64 and put the file on it. Use **Droid64** for this purpose. If you'd like to stick with DOS, use **SC** found in tools. After you've put the file on disk, open your emulator of choice (we recommend **VICE**). The first thing to do is to rename database the file. The C64 has a different logic when it comes to naming. Your database, currently PART1 needs to be renamed to [B]PART1, where [B] is an inverted B actually and PART is written with the shift key pressed so that the output will be symbols and no letters. This is how the whole command looks like from Commodore Basic:

12

If you have that, you have it right. Note there is a similar file on the disk image, which is the graphics database. You recognize it because it shows up with an inverted A in the directory. You need to leave that on the image, even if you create a text-only adventure as the file contains the charset for your game. If you want to use a different charset, there is one ripped from an Aventuras AD game on disk. Enter the DG editor, select the load charset option, provide the name NEWCHRSET when you're asked for a name. Save the graphics database as PART1 on disk again. Add graphics of course if you want your adventure to include graphics. DAAD has also facilities to make you import your graphics from the Spectrum graphics database. There are two interpreters on disk, EDI (English) and SDI (Spanish). We added loaders for English and Spanish games. So you can conveniently rename a loader. LE = Loader English and LS = Loader Spanish. LE1 for example will load an English game PART1. LS2 will load a Spanish game PART2. Finally delete from disk what you don't need anymore. This finally brings a sophisticated parser on par with PAWs to the Commodore 64. Enjoy! If you fancy adding a loader image to your adventure, have a look at the next section of this document, where the process is explained in detail. The information you've read here is the base of using the DAAD loader template supporting HiRes and Koala images.

Using the 8-bit DAAD loader templates for DISK/TAPE

Out of the box, it is not easy to add a loading screen to a DAAD 8-bit adventure. We wanted to ease the process for you and created templates you may now adopt with your own adventure games for that extra bit of professional look and feel. Use **Multipaint** as advised earlier in this documentation for creating your C64, ZX Spectrum, CPC or MSX loading screens and use **GrafX2** for creating your screens for Amiga, Atari ST and DOS. There is also **png2scr.py** which could create a Speccy screen from a given PNG image for you.

Spectrum: TAPE MASTER

Put your Spectrum binary (or binaries) into the TAPEMAST directory. You've created said binaries when following the "preparing a Spectrum tape release" instructions earlier in this documentation. Now put your Spectrum loading screen as **SCREEN.SCR** into the directory. Make sure it has a Spectrum header. Multipaint SCR files for example don't have one but you can add it simply by using the **ASH** tool. Rename your binary, which may be called PART1 for example, to **GAME**. In DOSBox, go to the TAPEMAST directory and enter the following command:

tapcat -ftap sidea.tap -b10 SPECCY.BAS SCREEN.SCR GAME


Repeat these steps with a possible PART2 binary but name the output file **sideb.tap**, so that the already existing file won't be overwritten. That's it, your Spectrum tape is mastered and ready for distribution. You can use **tape2wav** from the fuse-emulator-utilities in case you also want to create a WAV file.

Spectrum: DISK MASTER

Create a copy of the DAAD_Sloader.dsk image in LIB\SPECTRUM. Use CPCDiskXP to add your game databases (PART1.DDB, PART2.DDB) on the disk. Also add your loading screen as SCREEN.SCR. Make sure the .SCR file has a Spectrum header as described in the Spectrum: TAPE section. Rename DAADLOAD to the name of your game, SHERLOCK for example. Done.

In case you just have a one part adventure, use the MERGE command from +3 BASIC to modify DAADLOAD so it just loads part 1 instead of prompting which part to load. In that case, the PART2.SDG can be deleted, too. Note the template is made for English games but you can

Amstrad CPC: GENERAL



```
      DAAD Adventure Loader
(C) 2019 Stefan Vogt, Pond Soft
      https://8bitgames.itch.io

      Loading part 1...
```

The loaders for CPC are special as they do not just load an adventure with a loading screen. They also patch the game in memory before it gets executed. DAAD on CPC had two issues. You can only save to drive A: (when saving to disk) and a DAAD adventure won't run on a French AZERTY CPC, which is a major problem (the M key can't be pressed). The loader now takes care of the issues. So when using the DAAD CPC Disk loader you can save to any drive you've started the game from and your game will run just fine on a French AZERTY CPC, both from tape and disk. Please note that these loaders very likely only work with the English interpreters. One could delete the machine code patch (pretty much) at the end of the loader code though to make it work. We guess making a Spanish adventure run on a French machine is a pretty uncommon scenario anyway. But just for the loader itself it is definitely possible to adopt it for Spanish games. Just use the MERGE command from CPC BASIC and alter as needed so it fits your game.

Amstrad CPC: CREATING THE .SCR FILE

Note that DAAD uses a special format that appends the palette bytes at the end of the screen data, so normal SCR files won't work unless the palette is not appended. We added the **MARS** utility to tools, which does the job.

This is the recommended way: use Multipaint to create a CPC mode 0 image, save as PNG, you might need a separate image editor like GIMP to index it afterwards. We would recommend **png2crtc** (not bundled with DAAD) to create a .SCR file from your indexed PNG image:

png2crtc MYSCREEN.PNG SCREEN.SCR 7 0

Now dump the palette using **dump-pal.py** found in the TAPEMAST directory:

dump-pal.py MYSCREEN.PNG COLOUR.PAL

Use the **MARS** tool to merge SCREEN.SCR and COLOUR.PAL to **LOADING.SCR**, which is the format and the name the CPC loader needs. Use AAH to add an Amstrad header if needed (can also be done with CPCDiskXP).

Amstrad CPC: TAPE MASTER

Put your LOADING.SCR into the TAPEMAST directory. Also add your game binaries named PART1.BIN and PART2.BIN. Now use 2CDT to create the tape masters, replace GAMENAME with a short equivalent to your game's name, QUEST for example:

PART1:

```
2cdt -s 1 -n -r GAMENAME DLPART1.BAS side1.cdt
2cdt -s 1 -r DAAD.FNT DAAD.FNT side1.cdt
2cdt -s 1 -F 2 -L 0xc000 -r LOADING.SCR LOADING.SCR side1.cdt
2cdt -s 1 -r PART1.BIN PART1.BIN side1.cdt
```

PART2:

```
2cdt -s 1 -n -r GAMENAME DLPART2.BAS side2.cdt
2cdt -s 1 -r DAAD.FNT DAAD.FNT side2.cdt
2cdt -s 1 -F 2 -L 0xc000 -r LOADING.SCR LOADING.SCR side2.cdt
2cdt -s 1 -r PART2.BIN PART2.BIN side2.cdt
```

That's it. You may once again use tape2wav from the fuse-utilities to create a WAV file if you have the desire to get this on real tape, too.

Amstrad CPC: DISK MASTER

This one is easy. Make a copy of the DAAD_Loader.dsk image in LIB/CPC. Put your game binaries as PART1.BIN and PART2.BIN on it. Add your LOADING.SCR. Done. Use MERGE from CPC Basic to edit "DISC" in case you just have a one part game.

CBM 64

The C64 loaders support both HiRes and Multicolour images. HiRes is something you only see from the best artists in the scene, so when you're new to pixel artworks on the Commodore, we suggest to stick with Multicolour first. From Multipaint, save your work as a PNG image, then use a tool like Pixel Polizei or Pixcen to save it to the desired format. Rename your image to **LOADERPIC**. Make a copy of the DAAD Loader.d64 image in LIB/C64. Put your LOADERPIC on it. When using DROID64, the PRG suffix is added automatically. Now add your game databases and rename it using the naming convention you learned in the worked example section, earlier in this documentation. DLKOALA1 will load a Koala artwork before it launches PART1 of your adventure. DLKOALA2 will do the same with PART2. The DLHIRES1 and DLHIRES2 are the loader equivalents for HiRes images. Delete everything from disk you don't need anymore. Rename the remaining loaders to something that represents your game better, e.g. DLKOALA1 to QUEST1 or whatever. That's it, done.

Note that there is only EDI on the loader disk, the English interpreter. The quickest way to create a Spanish release would be to delete EDI, put SDI instead on disk and then rename it to EDI, since the loaders search for EDI

Known issues

Spectrum +3 games can only save to tape

Saving to disk had never been implemented to the DAAD Spectrum interpreters. So the game works, but it will ask you for a tape when you type the SAVE command. Tim is aware of the issue and said that we might be able to backport the feature from Gilsoft PAWs (+3 version), which can in fact save to tape and disk. Since PAWs and the DAAD interpreter are not too far away from each other, we are confident that you can expect an update in the future. Tim definitely will have a look. Stay tuned.

Credits

Some of the software we bundled with DAAD is not made by us. The project also received amazing contributions by even more amazing individuals. We want to give credit to the authors and advise you to support them in any way you can.

TAPCAT	→ Written by John Elliot as part of TAPTOOLS. You may find other programs in the TAPTOOLS bundle interesting.
SC / StarCommander	→ Written by Joe Forster. Please be so kind and register the program if you continue using it. It's giftware but we are sure Joe would appreciate that a lot.
2CDT	→ Written by Kevin Thacker.
C64 HiRes Loader	→ Written by Vanja Utne, based on Stefan's Koala loader

CPC Loader code
DAAD AZERTY patch
DAAD drive patch

→ All written by Nich Campbell - cpcgamereviews.com

dump-pal.py
png2scr.py

→ All written by Juan J. Martinez - usebox.net

Legal notes

The binaries of DAAD had been kindly gifted to the public domain by **Andrés Samudio**, the founder of Aventuras AD. It was his company that held the single right to use this software. The sources of DAAD's tools and the compiler are still copyright Tim Gilberts and Infinite Imaginations and are not distributed.

The source code of Hibernated 1 – This Place is Death is copyright Stefan Vogt and Pond Software. Read the license details in the header of the SCE file to learn more. Neither the name of the author nor the names of other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided “as is” And any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the author or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services loss of use, data, or profits or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

You are responsible for any legal issues arising from your use of this software.

Final words

It took us many hours, tea and fairy dust to craft this new DAAD release, all for the sake of preservation. DAAD is the final and most sophisticated tool emerging from the code that once started as “The Quill” on a rubber key ZX Spectrum. Now complete again for the first time in nearly 30 years, it never was available to the public and never to an English language audience. DAAD is a significant milestone in text adventure history, a heritage that is reflected by the wonderful “aventuras conversacional” from Aventuras AD. But rather than a “thank you”, we want you to use the system. Far too few adventures were written with it and the time couldn't be better to change that. Imagine worlds!

Tim and Stefan,
March 2019