DAAD
Adventure Writer

Version 2 - Release 2

A multi-machine adventure writing system.

Revised in September 2018

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 year 2018 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. With some efforts, you could use such a file though to port your adventure to DAAD. The sections that differ the most are objects, the process and response tables. DAAD also has no facility that automatically decrements integers (flags). If you used the auto-decrement logic in the past, you need to implement that yourself in DAAD.

System requirements and recommendations

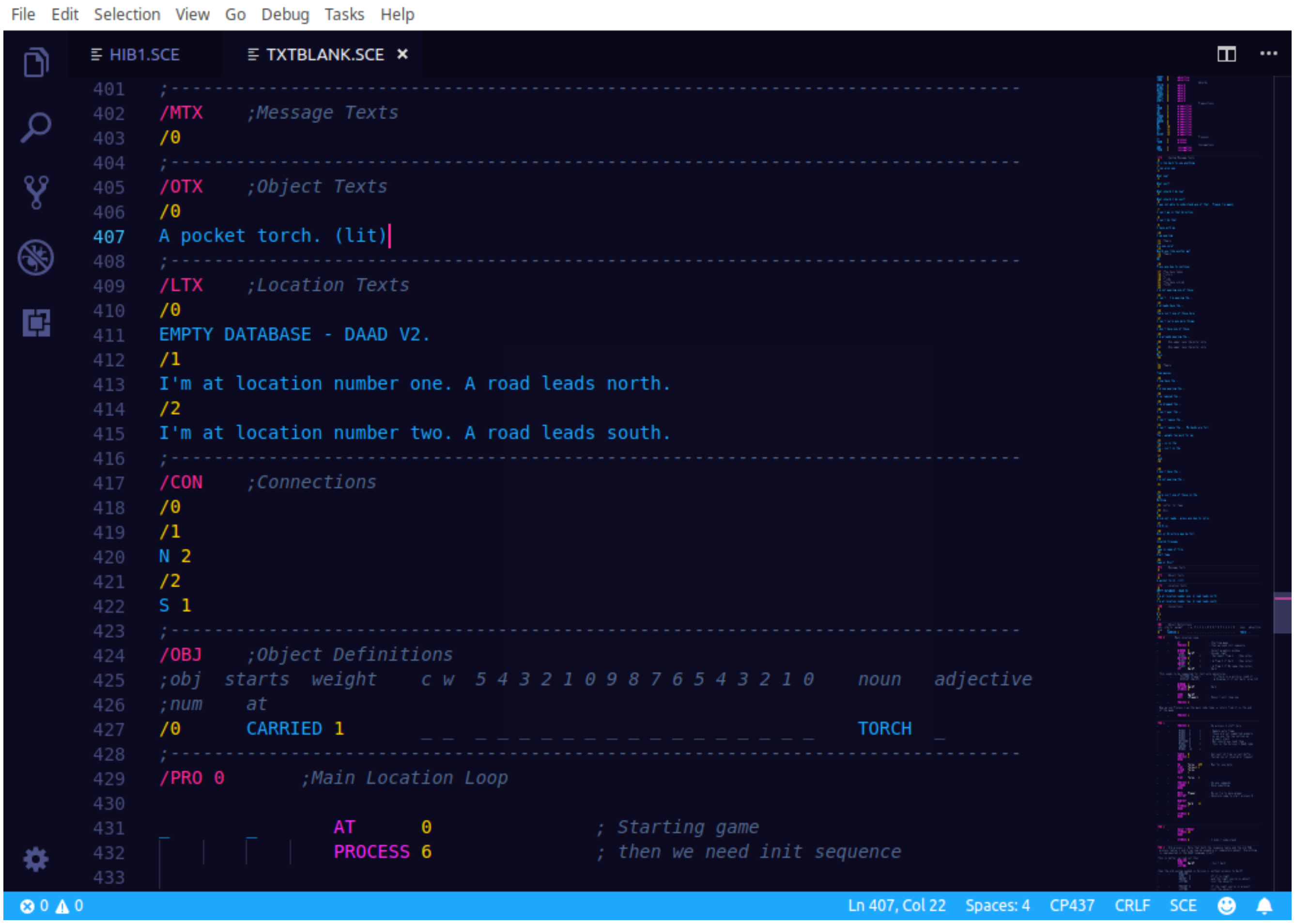
We wanted to ensure that you can work with DAAD in a (mostly) modern environment. Here are the mandatory system requirements to start developing your own adventures:

- 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:

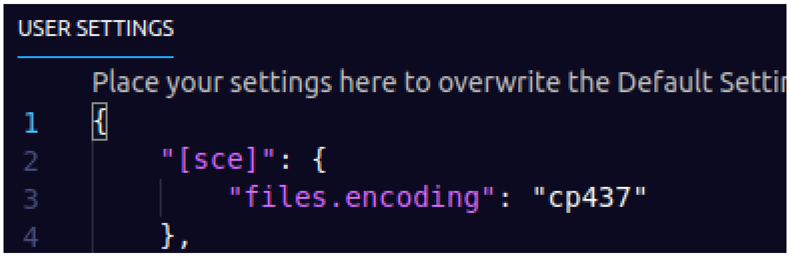

File Edit Selection View Go Debug Tasks Help
HIB1.SCE
TXTBLANK.SCE ×
0 0
Ln 407, Col 22 Spaces: 4 CP437 CRLF SCE

```
;------------------------------------------------------------------------------
/MTX    ;Message Texts
/0
;------------------------------------------------------------------------------
/OTX    ;Object Texts
/0
A pocket torch. (lit)
;------------------------------------------------------------------------------
/LTX    ;Location Texts
/0
EMPTY DATABASE - DAAD V2.
/1
I'm at location number one. A road leads north.
/2
I'm at location number two. A road leads south.
;------------------------------------------------------------------------------
/CON    ;Connections
/0
/1
N 2
/2
S 1
;------------------------------------------------------------------------------
/OBJ    ;Object Definitions
;obj  starts  weight    c w  5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0    noun   adjective
;num    at
/0      CARRIED 1       _ _  _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _    TORCH  _
;------------------------------------------------------------------------------
/PRO 0       ;Main Location Loop

_       _       AT      0               ; Starting game
                PROCESS 6               ; then we need init sequence

```

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 Setti

```
{
    "[sce]": {
        "files.encoding": "cp437"
    },
```

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 since DAAD v2 Release 1 (1991)

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 full-featured tape loader with SCREEN$

in **LIB\ST\**

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

Directory structure DAAD v2 R2 (2018)

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) 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

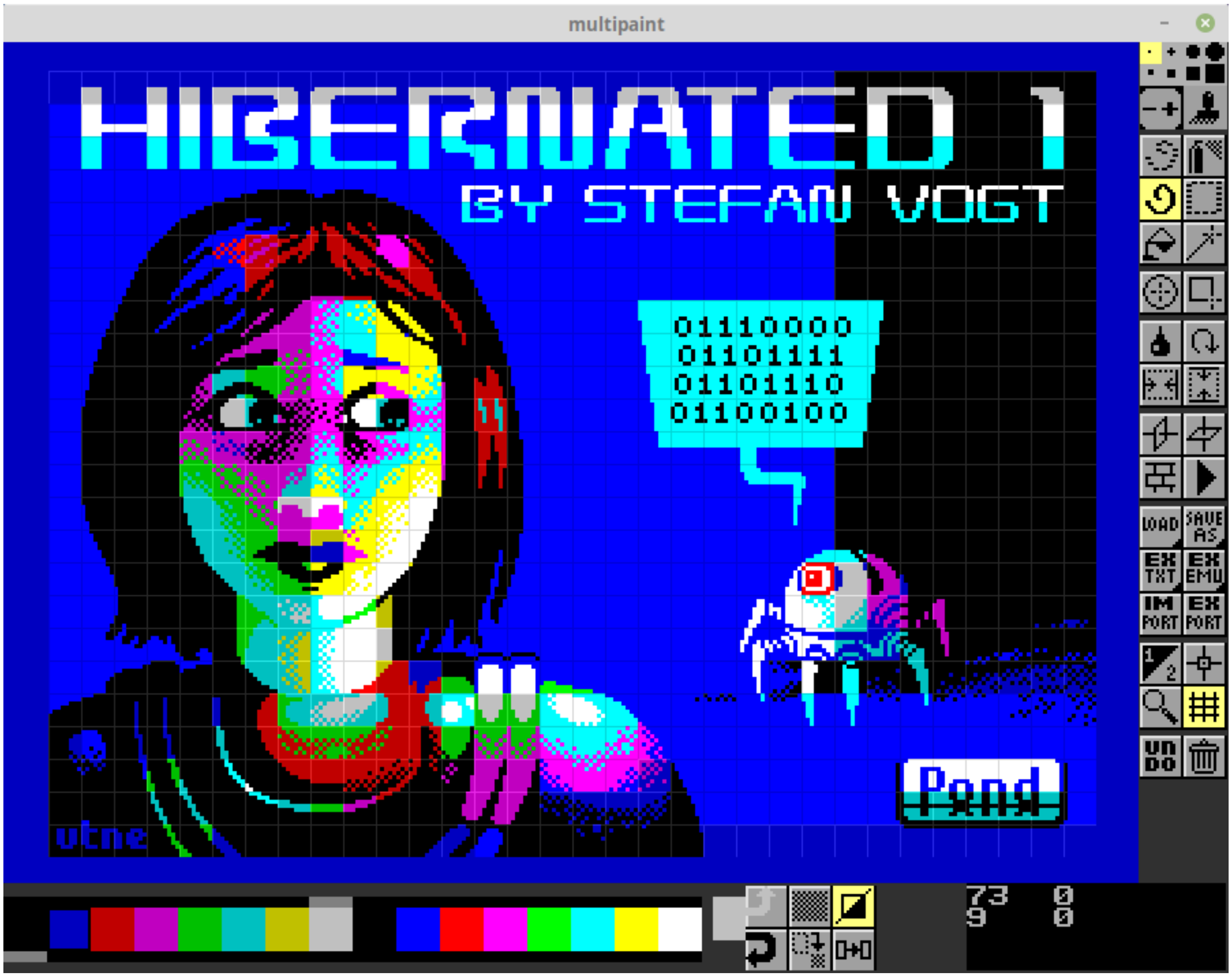
DAAD\OBJ\ → as referenced in the 1991 documentation, no changes

DAAD\INTERP\ → contains all the DOS interpreter files, see 1991 docs

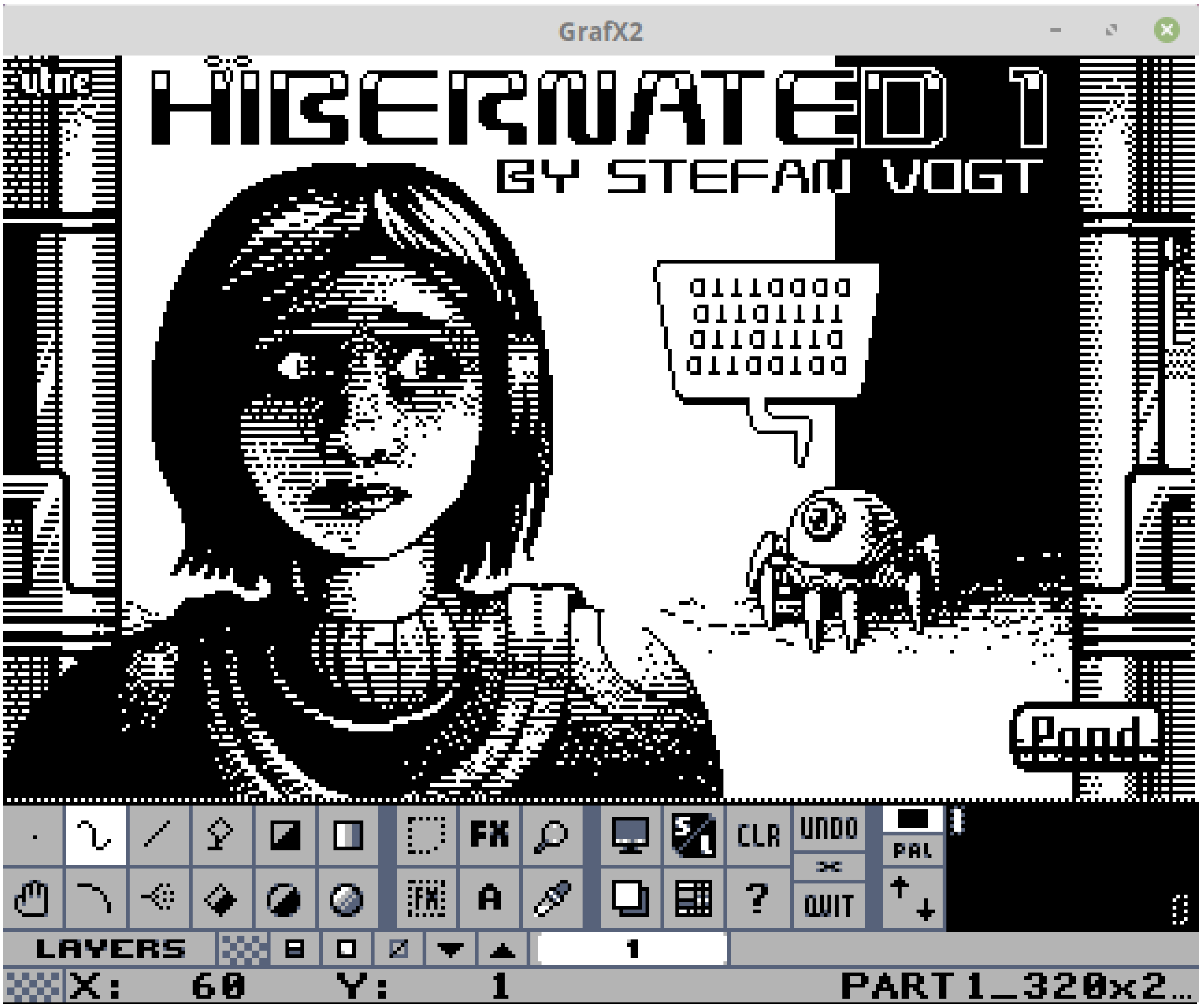
LIB → interpreters and tools to roll out your games on the supported platforms

DOCS → the documentation

Image editing – pixel graphics and loading screens



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.



A worked example in modern times

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.

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 **WinAPE** to transfer your game files to an Amstrad disk. WinAPE has a “disk edit” mode, conveniently allowing to alter contents via drag and drop. WinAPE is a Windows application but it will run with Wine under Linux. On Linux we recommend **Arnold** for all tasks other than moving bits and bytes around. 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 oufile{.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 you could use WinAPE, on Linux we use a terminal utility called 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 Amstrad emulator to move files to a Spectrum +3 disk image. Both machines use the same disk system so it will work. Close WinAPE 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.

ZX Spectrum: creating a Spectrum +3 release
Open +3 basic. Use one of the +3 Basic loader templates to create a loader for your game with the command **MERGE "DRE"**. Note that it's "DRS" for Spanish adventures. Delete the line where it loads a screen in case you don't provide one. Change the name of your loader screen if it differs. Now save the loader to disk with the command **SAVE "MYGAME" LINE 10**. Replace MYGAME with the actual name you want to give your loader. Now delete everything from disk that was not referenced in your loader file. Deleting can also be done from WinAPE. That's it.

ZX Spectrum: creating a tape release (.TAP file)
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 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:

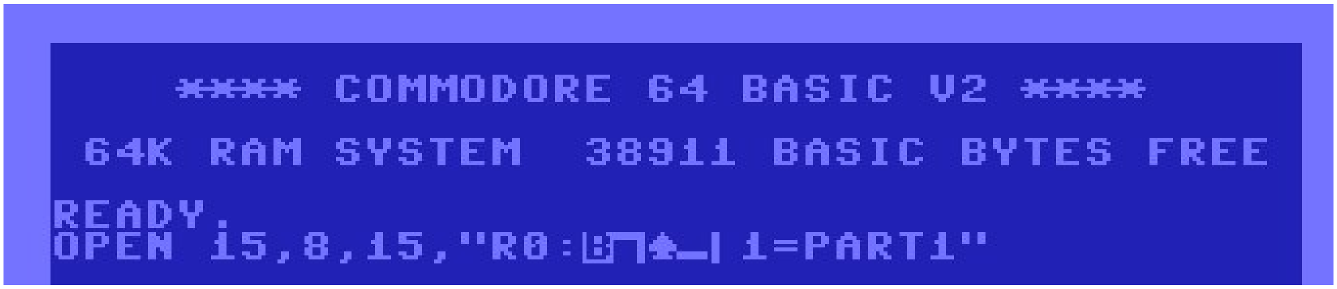
SAVE "MYFILE" CODE 24576,40960

Replace MYFILE with your game's name of course. You also need a tape loader. There is a file TAPLOAD on disk, which can be used as a full-featured template. Use the MERGE command again to have a look at it and maybe delete the line where it loads a screen in case you're not providing one. Save it to disk via **SAVE "TLOADER" LINE 10** or just overwrite it. If you choose to overwrite it, the system will create a .BAK file anyway. Now you need to get your loader and the game binary from the +3 disk image into the DAAD structure in DOSBox. You could use WinAPE again for this purpose. IDSK will do the job, too. Put the two files into the tape mastering directory at DAAD\TAPMAST. Finally use **TAPCAT** to create the final TAP file. TAPCAT is a DOS utility, you don't need to switch to a Windows command shell for that one. A short documentation containing a working example for the constellation of the DAAD TLOADER script together with a binary can be found in the tape mastering directory, too. Congratulations, tape ready for distribution!

CBM 64

The Commodore 64 was our problem child due to corrupted disks, outdated interpreters, missing headers. We are going to replace the 1991 howto completely with this one.

Compile a version of your source without debug information compressed called PART1.DDB (etc) using option -m2c. 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. You can either use SC from the tool collection for this purpose, you may also use an alternate tool for handling Commodore 64 disk images. A good one that is cross-platform is DROID64, which is working fine on Linux. 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:



```
    **** COMMODORE 64 BASIC V2 ****

 64K RAM SYSTEM  38911 BASIC BYTES FREE

READY.
OPEN 15,8,15,"R0:▣┐♠▁│1=PART1"
```

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!

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 near future. Tim definitely will have a look. Stay tuned.

French AZERTY CPCs are not supported

Sadly, CPC games won't work on French CPCs with AZERTY keyboard. The "M" key can't be pressed. We weren't even aware those machines exist and we actually have no idea what Amstrad thought they were doing. It turns out these units suffered a lot from incompatibility in the past. Many games had to be altered for the French market. The keymaps are hardcoded into the interpreter so this issue is hard to resolve. The code remains untouched since 1991. We want to be honest. At the moment, we are not confident getting the games run on AZERTY units. These machines work fine: CPCs from UK, Spain, Italy, Germany (Schneider), basically all except French units with AZERTY keyboards. The older French units have QWERTY keyboards and will work.

Credits

Some of the software we now bundled with DAAD is not made by us. We want to give credit to the original authors and advise you to support them in any way you can. SC for example, even though it's giftware, may be registered.

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. We are sure Joe would appreciate that a lot.

2CDT → Written by Kevin Thacker.

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.

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,
September 2018