

in

Ben Hutchings

COLLABORATORS

	<i>TITLE :</i> in		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Ben Hutchings	February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	in	1
1.1	Contents	1
1.2	Legal Information	1
1.3	distribution	2
1.4	disclaimer	2
1.5	Third Party Copyright	3
1.6	Introduction	3
1.7	ModUtils History	3
1.8	About Modules	4
1.9	Installation	5
1.10	reqtools	5
1.11	modplay	6
1.12	Using ModPlay in Scripts	8
1.13	ModPlay History	8
1.14	modrip	10
1.15	ModRip History	11
1.16	modalloc	11
1.17	ModAlloc History	12
1.18	makeripperdisk	13
1.19	modwipe	13
1.20	Acknowledgements	14
1.21	How and Why to Contact Me	14

Distribution

Disclaimer

Third Party Copyright

All trademarks referred to herein are the properties of their respective owners.

1.3 distribution

The contents of the 'ModUtils' package are the files:

Install ModUtils	Install ModUtils.info
Install ReqTools	Install ReqTools.info
MakeRipperDisk	MakeRipperDisk.info
ModAlloc	ModAlloc.info
ModPlay	ModPlay.info
ModRip	ModRip.info
ModUtils.doc	ModUtils.doc.info
ModUtils.guide	ModUtils.guide.info
ModWipe	ModWipe.info

These files are all copyright © Ben Hutchings 1995. They may freely be distributed provided

(i)~~~no charge is made for this above the costs of duplication, distribution and the media used,
(ii)~~all the aforementioned files are distributed together, and
(iii)~none of the aforementioned files are altered, except, if the distributor so wishes, by converting them into a compressed form from which they can be retrieved unaltered.

Software released under these conditions is often known as Freeware.

The file '

reqtools.library

' is intended to be included with this package

but I do not hold the copyright to this file and so I have no right nor intention to restrict its use or distribution.

1.4 disclaimer

This software and documentation is provided 'as-is' without representation or warranty of any kind, either express or implied, including without limitation, any representations or endorsements regarding the use of, the results of, or the performance of the information, its appropriateness, accuracy, reliability, or currentness; the entire risk as to the use of this information is assumed by the user.

In no event shall I be liable for any damages, direct, indirect, incidental or consequential, resulting from any defect in the information, even if I have been advised of the possibility of such damages.

1.5 Third Party Copyright

I neither condone nor authorise the use of my software, specifically, the program 'ModRip', for the illegal reproduction of copyright material. It is illegal to make copies, even for your own use, of copyright material, without the express permission of the copyright owner, or as permitted by the license thereof.

1.6 Introduction

If you aren't familiar with music modules, you should first ↔ read the section which explains a bit about them.

If you are familiar with the ModUtils package, you may want to read the release history for a summary of changes.

The ModUtils package consists of a module player -
ModPlay
- and a module
ripper -
ModRip
- plus a few extra bits of software to support them.

I wrote these programs after finding my existing module software inadequate and out-dated. I hope you find them as useful as I have. If you find these inadequate or out-dated then I'm very sorry!

1.7 ModUtils History

release 1.27, 7 February 1996

ModPlay 2.10, ModRip 1.1, ModAlloc 1.3, MakeRipperDisk 1.0, ModWipe 1.0

release 1.26, 25 January 1996

ModPlay 2.9, ModRip 1.1, ModAlloc 1.3, MakeRipperDisk 1.0, ModWipe 1.0

release 1.25, 14 August 1995

ModPlay 2.8, ModRip 1.1, ModAlloc 1.3, MakeRipperDisk 1.0, ModWipe 1.0
release 1.24, 1 August 1995

ModPlay 2.7, ModRip 1.1, ModAlloc 1.3, MakeRipperDisk 1.0, ModWipe 1.0
You may notice that ModAlloc seems to have jumped straight from v1.0
to v1.3. This is because the version included in previous releases was
actually v1.2 (see ModAlloc history).
release 1.23, July 1995

ModPlay 2.7, ModRip 1.1, ModAlloc 1.2, MakeRipperDisk 1.0, ModWipe 1.0
release 1.22, June 1995

ModPlay 2.6, ModRip 1.1, ModAlloc 1.2, MakeRipperDisk 1.0, ModWipe 1.0
release 1.21, May 1995

ModPlay 2.5, ModRip 1.1, ModAlloc 1.2, MakeRipperDisk 1.0, ModWipe 1.0
~Improved documentation, thanks partly to a text formatting program I
just wrote.
release 1.2, May 1995

ModPlay 2.4, ModRip 1.1, ModAlloc 1.2, MakeRipperDisk 1.0, ModWipe 1.0
release 1.1, November 1994

ModPlay 2.3 & 1.0, ModRip 1.0, ModAlloc 1.2, InstallBB 1.0, ModWipe
1.0
release 1.0, October 1994

ModPlay 2.2 & 1.0, ModRip 1.0, ModAlloc 1.2, InstallBB 1.0, ModWipe
1.0

1.8 About Modules

Music modules are files containing descriptions of one or more sounds and a list of notes and effects to play with them. One of the earliest programs to produce music modules was Soundtracker, which produced simple files containing sampled waveforms (that is, digital records of real instruments and noises) plus a number of blocks of note codes, and a list defining the order they should be played in (so that blocks could be repeated).

Then a program called Noisetracker was released which had added features and used roughly the same file format, so that it could read in modules produced by Soundtracker. Since then numerous improved versions have been produced and Soundtracker has been replaced by Protracker. Despite all these changes the file format has remained the same except that extra codes for the new musical effects have been added. So there are huge numbers of modules out there which can all be handled by the same player

program. Not only this, but the same file format is also supported by software on the PC and Archimedes, and maybe some other computers. A friend of mine has hundreds of modules of this type on PC CD-ROM. Using DOS2DOS, MessyDOS, or CrossDOS then you will be able to use these modules as well as those on normal Amiga disks!

Many demos and games use Soundtracker or compatible modules because it is much easier to use a standard program to make music than to write a special routine for each tune. So if you can get the module out of the game (a process known as 'ripping') then you can listen to your favourite game tunes while doing something else like word-processing or writing a program. However, please note the legal implications of this.

The limitation of ripping is this: unless you have an Action Replay or similar cartridge, you can't actually stop the game while it's running and rip out data without disturbing anything. Nevertheless, I have found my ripper software to be very effective.

Included in this package are a module player -
 ModPlay
 - and a module
 ripper -
 ModRip
 - plus a few extra bits of software to support them.

1.9 Installation

You do not need to install any of the programs in order to use them, but you may find it useful to do so if you have a hard drive.

If you have Commodore's Installer program, all you need to do is to double-click the 'Install ModUtils' icon then follow the instructions you are given. If you aren't sure whether you have the program, there's no harm in trying this anyway.

If you don't have Installer, then you will have to copy the programs to your hard drive by dragging their icons or by using the Shell. (To install all the programs you can simply drag the drawer icon.) If you install ModPlay individually, manually, please refer to the section concerning

reqtools.library
 , which is required to provide full functionality.

1.10 reqtools

Reqtools.library is a set of programmers' routines created by Nico François for use on the Amiga. ModPlay uses it to create file requesters

and error requesters, so you should have it installed on your system before using ModPlay.

If you use the 'Install ModUtils' icon to install ModPlay then this will be done automatically. If you carry out a manual installation, you should then check whether you have a copy of reqtools.library by running the installed copy of ModPlay from Workbench to see whether it works. If it produces an error, then you should install ReqTools on your system disk or partition by double-clicking the 'Install ReqTools' icon on the Workbench.

Note that reqtools.library is copyright © Nico François.

1.11 modplay

Those familiar with ModPlay might want to read the release history for a summary of changes.

ModPlay is a simple module player based on the Protracker playroutine. The advantages it has over other module players are these:

- It's very small.
- It can be made resident in memory for Shell use.
- It can stop music automatically instead of looping.
- It uses

ReqTools
(thanks to Nico François).

- It uses reliable CIA timing so the speed is correct whatever screenmode you are using.
- You can stop or change the music being played simply by running ModPlay again.

One of the reasons I had for writing this was for use in disk magazines (I was hoping to produce one myself but that will have to wait). Although it can be run from Workbench, ModPlay is designed for control by Shell commands and especially by scripts.

Note that you must have reqtools.library version 38 or better in your LIBS: directory if you want to run it from the Workbench or use a file requester to select music modules.

ModPlay can automatically detect and decompress modules packed with the PowerPacker routines, but to do so it requires "powerpacker.library" to be present in the LIBS: directory. This can be found in many PD/freeware distributions, on coverdisks, and in the full PowerPacker package, or via the Internet in Aminet archive util/libs/pplib16.lha.

Shell usage:

For full functionality from a Shell under 1.3, ModPlay requires the arp.library. If arp.library is not available, ModPlay will only play one named module or display a requester. All text following the command name and the space after it will be regarded as a filename, though for the sake of compatibility any quotes around the filename will be ignored. If no arguments are given a requester will appear.

To start playing a module, or to change some music already playing, type

```
ModPlay <file> [LOOP]
```

where <file> represents the filename of the module, and LOOP is an optional switch. Multiple files can be supplied, and they will be played in the order specified. If a single file is specified and the LOOP switch is used, the file will be played in a loop until it is stopped manually. If the LOOP switch is used with more than one file, ModPlay will run through the sequence of files repeatedly until it is stopped manually.

To stop the music prematurely, or to stop a looping piece of music, type

```
ModPlay STOP
```

To use a file requester to start/stop music, just type

```
ModPlay
```

then pick a module or click Cancel to stop. The requester will keep coming back when each piece of music finishes, until you click Cancel.

You can also stop a single module by pressing Ctrl-C, or you can completely exit the program by pressing Ctrl-D.

If you intend to use ModPlay in a script, please see the separate note concerning this.

Workbench usage:

There are three possible ways to start ModPlay from the Workbench.

- 1.~If your modules are saved with icons, you can set their default tool (using the Workbench 'Information' function) to something like 'Work:ModUtils/ModPlay' (this setting would be correct if you have installed ModUtils on a partition called 'Work'). Then, when you double-click them, ModPlay will be loaded and will then play them.
- 2.~If your modules have icons, you can use extended selection to play them. What this means is, select the first one, then hold down Shift and select any others, then (keeping Shift held down) double-click ModPlay's icon.
- 3.~Double-click the ModPlay icon. A file requester will appear. Select the drawer your modules are stored in. Then select a module and click OK. The requester will re-appear when the module(s) stop. You can then stop by clicking Cancel or select a module or modules again to go on playing.

You can stop a module before its natural end by starting ModPlay from its

icon and clicking Cancel.

1.12 Using ModPlay in Scripts

If you use ModPlay in a script for a disk magazine or similar, you will need to run ModPlay in 'background' with the Run command. If you want to put something like

```
ModPlay intro.mod
```

in a startup-sequence, e.g.

```
Echo "Welcome to XYZ disk magazine issue #5!"  
ModPlay intro.mod  
LoadWB  
EndCLI >NIL:
```

you will actually need to use the command

```
Run >NIL: <NIL: ModPlay >NIL: <NIL: intro.mod
```

otherwise the Shell window will just stay on the screen and do nothing until the music finishes.

If ModPlay is asked to STOP and there isn't actually any music playing, it will return code 5 (WARN) in case you want your script to check for this. Obviously if something else goes wrong it will return code 20 (FAIL) which will terminate your script unless you use the Failat command (see your Commodore manual for details) to prevent this.

1.13 ModPlay History

version 2.10, 7 February 1996

- ~Got rid of the irritating high-pitched noises which sometimes got played when a sample ended. These appeared when playing non-looping samples which didn't have zero in their first two bytes, because 'non-looping' samples are actually set to have loop start 0 and loop length 2 bytes. Now a dummy zero sample will play as the loop part of such samples.

version 2.9, 25 January 1996

- ~PowerPacker decompression added.
- ~Error reporting improved and debugged (I hope).
- ~More tolerant of modules with incorrect lengths, as produced by a number of Trackers.

version 2.8, 14 August 1995

- ~Double-quotes are now handled consistently even in the absence of OS
-

2.0+ and arp.library. I felt that the approach introduced in v2.4 seriously damaged the program's compatibility in automated systems e.g. scripts.

- ~Really fixed library opening - now the LIBS: directory is looked in first, followed by the current directory (CLI) or tool directory (Workbench).

version 2.7, 31 July 1995

- ~There was a really stupid bug introduced in v2.5 which meant that if you started one copy of ModPlay with a requester, then tried to play something with a second copy, not only would the second copy stop the first (as expected) but the first would then stop the second (whoops!) Fixed.

version 2.6, 16 June 1995

- ~Fixed a couple of bugs I rather stupidly introduced into the playroutine. Songs with arpeggios in will no longer crash now!
- ~If the loop switch is used with multiple files, the sequence of files will be looped rather than each individual file.
- ~Fixed a small bug in the library opening which probably never caused any trouble.
- ~I got fed up with the pickiness over module size, so now any extra bytes on the end of a module will just be ignored.

version 2.5, 30 May 1995

- ~Put back the code to open libraries in the current directory (as used in version 1.0).
- ~If ARP command line parsing fails, the ARP error message is now used instead of unhelpfully reporting 'unknown error'.
- ~Rewrote some of the C code in assembler to save on code size, and removed a few bits of code which never actually got called. Also optimised the playroutine a bit more.
- ~Requester now no longer re-appears if instructed to stop by another copy of ModPlay (this bug only appeared in v2.4!)
- ~Under OS 1.x you can now play one module or use a requester from the CLI without arp.library (as was possible before v2.4!)
- ~A potentially nasty bug introduced into the playroutine has been removed. I don't think it would ever have caused any problems though.

version 2.4, 11 May 1995

- Unified the two versions for OS 1.x and OS 2.0+, so all the improvements made since version 1.0 can now be enjoyed by everyone. Actually, version 1.0, which was first released as ModPlay13, didn't work at all (reported by Alan Pfeil). Strangely, the preliminary v1.0 I made two days earlier, which I myself still had installed, did. I can't find an explanation for any of this strangeness.
 - Arp.library is now required for Shell use under OS 1.x (but see note for v2.5)
 - Reqtools.library is not opened unless and until any requesters are required (suggested by Alan Pfeil). This means that if no user intervention is required (e.g. you are putting ModPlay in a script on a self-booting disk), reqtools.library does not need to be available.
-

version 2.3, November 1994

- Fixed a bug in the code for changing directories - the program often (usually?) failed inexplicably if started from the Workbench.

version 2.2, September 1994

First public release.

versions 2.0-2.1, 1994

Not released.

version 1.0, 1993

Released along with v2.2 in ModUtils r1.0, as ModPlay13.

1.14 modrip

Those familiar with ModRip might want to read the release history for a summary of changes.

ModRip is a simple module ripper. This means that it is designed to identify any Soundtracker or compatible modules in memory, calculate their size, and save them to disk.

Despite this apparently rather dodgy mode of operation, it should be completely system-legal as it only reads memory within the bounds of the memory blocks listed as public memory. This will cause problems if you try running the program 'Mungwall' at the same time. It may also cause problems with 'Enforcer' but I don't know since I'm not quite sure what Enforcer does. If so, please tell me.

It isn't 100% reliable as most games and demos bypass the operating system and so are liable to have their data corrupted (i.e. destroyed) in the period between exiting the program and running ModRip. Also some games intercept resets and clear their data in memory to prevent ripping. Besides which, not all game music will be stored in Soundtracker format. Note that I have also produced a ripper for MED modules as part of my MEDUtils package, which may accompany this.

The effectiveness of ModRip can be improved using ModAlloc or MakeRipperDisk.

Please note that ripping is not always

legal

.

Shell usage:

To begin searching for modules, type

ModRip

and wait. When a module is found, brief information about it will be displayed in the window. If this is gibberish then don't bother saving it - most modules contain some meaningful text which you will be able to read. To save the module, type in a filename - e.g. DF0:NiceRayTracedDemoTune - and press Return. To skip it, just press Return.

When the search is finished, a Shell prompt will reappear.

Workbench usage:

Double-click the ModRip icon.

When a module is found, brief information about it will be displayed in a window. If this is gibberish then don't bother saving it! Most modules contain meaningful text which you will be able to read. To save the module, type in a filename - e.g. DF0:NiceRayTracedDemoTune - and press Return. To skip it, just press Return.

When the search is finished, the window will disappear.

1.15 ModRip History

version 1.1, May 1995

- Although ModRip never used any OS 2.0 functions, the mode previously used to open an output window when ModRip is started from Workbench is not supported by OS 1.x! This made the window close immediately. This has been fixed.

version 1.0, August 1994

First public release.

1.16 modalloc

Those familiar with ModAlloc might want to read the release history for a summary of changes.

This program installs a special 'KickTag' which is run immediately after any reset. The KickTag code will find any Tracker modules in memory and allocate the memory they use so that they are protected from being written over. This means that you can rip many modules which you might otherwise not be able to.

To run it, simply double-click the ModAlloc icon or use the Shell. A message will appear telling you to reboot when ready. This means that you can now run whatever program/game/demo you want to rip music out of, then reset the computer when the music you want is playing. Once ModAlloc has been run, every time you reset, the screen will start to strobe slowly. At this point you should press the LEFT mouse button to start searching for modules or the RIGHT mouse button to bypass this and reboot normally.

If you press the left button, the screen will immediately turn red. If and when a module is found it will turn blue until it finds the end of the module, when it will briefly flash yellow. This may happen several times. Be patient! The process should take a maximum of 5 seconds for each megabyte of memory switched on; if it takes much longer than this, you can press the RIGHT mouse button to exit. Once the procedure has finished or you press the right mouse button to exit, the computer will boot up as normal. You can then run ModRip to rip the modules as described above.

Note that some programs may completely tear down the operating system, which means the 'KickTag' code will no longer be found and will not operate when you reset. In this case you should use
MakeRipperDisk
.

1.17 ModAlloc History

version 1.3, August 1995

..It works! And it's also faster and more reliable than the original version.

version 1.2, August 1994

..Erm, don't know what improvements I was trying to make here. This was the first released version. Unfortunately, it didn't work.

version 1.1, July 1994

..Some improvements on v1.0, I guess. I can't remember that far back.

version 1.0, July 1994

First version.

1.18 makeripperdisk

This program makes a special bootable disk which should provide the best environment for ripping modules. ←

Just find a blank disk (no need to format it) and run the 'MakeRipperDisk' program either from the Shell or by double-clicking its Workbench icon. Then, when you are asked to, insert the blank disk and press Return. Within about a minute your disk (which will now be called 'ModRip') will be ready. I should mention that the counter for the formatting procedure counts tracks and not cylinders, so it will count up to 159 on a double-density disk or 319 on a high-density disk. Do not be alarmed! Note that I cannot test the program with high-density disks so I do not know for sure whether it will work correctly with them; please tell me if it doesn't.

To use the ripper disk, first turn off all unneeded drives, then reset the computer and run the program/game/demo you want to rip music out of. Then reset again and quickly insert the ripper disk in the internal drive (do NOT wait for the nice picture of a disk to appear!). The screen will start to strobe slowly. At this point you should press the LEFT mouse button to start searching for modules. The screen will then turn red. If and when a module is found it will turn blue until it finds the end of the module, when it will briefly flash yellow. This may happen several times. Be patient!

The process should take a maximum of 5 seconds for each megabyte of memory switched on; if it takes much longer than this, you can press the RIGHT mouse button to exit. Once the procedure has finished or you press the right mouse button to exit, the computer will boot up as normal and

ModRip will load. There will be plenty of space on the disk to save any modules which are found. ←

1.19 modwipe

This tiny utility just wipes out any modules in memory so that they are invisible to ←

ModAlloc and the ripper disks made with MakeRipperDisk and won't take up any valuable memory. It will then reboot, freeing all memory and allowing you to restart with all your memory available.

It will take a little while before it resets to check all through memory - a maximum of 5 seconds for each megabyte of memory switched on. During this time nothing will seem to be happening - this is normal.

WARNING: Before running ModWipe you should ensure that any data you are working on is saved to disk, and that all disk access has stopped. Otherwise you could lose your work or worse still corrupt a disk!

Shell Usage:

ModWipe

Workbench usage:

Double-click the ModWipe icon.

1.20 Acknowledgements

I owe the main player routine mainly to Lars 'Zap' Hamre ←
and Amiga

Freelancers - and if you're out there somewhere, please, please, can I have a well-commented up-to-date routine because this one dates from 1991 and I had to spend months hacking at it just so that I could understand it, after which it needed a number of enhancements to handle the modules I threw at it. Mind you, I'm not much good at writing in assembly language anyway and custom hardware was a complete mystery to me before I read this code.

Thanks to Nico François for
retools.library
and its excellent
documentation, which is, like the library itself, freeware.

I must also thank Alan Pfeil of ICPUG for drawing my attention to a number of serious bugs in earlier versions of ModPlay.

//

Finally, thanks to \X/ Amiga for being the best.

1.21 How and Why to Contact Me

Please write to me if you have any comments, suggestions, bug reports or programming hints to make. I would especially like to hear from anyone who can supply

·~some 'official' tracker specifications

or~a new playroutine (I am currently using a modified version of Protracker 1.1B VBlank)

or~any reports of Enforcer hits (well I don't exactly want to hear these but if there is a problem I ought to know)

Snail mail:

Ben Hutchings
43 Harrison Close

Reigate
Surrey RH2 7HS
ENGLAND

E-mail (Internet): benjamin.hutchings@worc.ox.ac.uk

For up-to-date information on ModUtils and other software by me see my web
page: <http://sable.ox.ac.uk/~worc0223>
