

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	MEDUtils History	3
1.8	About Modules	4
1.9	Installation	4
1.10	reqtools	5
1.11	medplay	5
1.12	Using MEDPlay in Scripts	7
1.13	MEDPlay History	8
1.14	medrip	9
1.15	MEDRip History	10
1.16	medalloc	10
1.17	MEDAlloc History	11
1.18	makeripperdisk	12
1.19	medwipe	12
1.20	Acknowledgements	13
1.21	How and Why to Contact Me	13

Distribution

Disclaimer

Third Party Copyright

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

1.3 distribution

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

Install MEDUtils	Install MEDUtils.info
Install ReqTools	Install ReqTools.info
MakeRipperDisk	MakeRipperDisk.info
MEDAlloc	MEDAlloc.info
MEDPlay	MEDPlay.info
MEDRip	MEDRip.info
MEDUtils.doc	MEDUtils.doc.info
MEDUtils.guide	MEDUtils.guide.info
MEDWipe	MEDWipe.info

The player routine contained in MEDPlay is copyright © Teijo Kinnunen and RBF Software.

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 'MEDRip', 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 MEDUtils package, you may want to read the release history for a summary of changes.

The MEDUtils package consists of a module player -
MEDPlay
- and a module
ripper -
MEDRip
- plus a few extra bits of software to support them.

I wrote these programs after finding my existing module software inadequate. 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 MEDUtils History

release 1.22, 6 November 1995

MEDPlay 2.6, MEDRip 1.1, MEDAlloc 1.3, MakeRipperDisk 1.0, MEDWipe 1.1

release 1.21, 14 August 1995

MEDPlay 2.5, MEDRip 1.1, MEDAlloc 1.3, MakeRipperDisk 1.0, MEDWipe 1.1

release 1.2, 2 August 1995

MEDPlay 2.4, MEDRip 1.1, MEDAlloc 1.3, MakeRipperDisk 1.0, MEDWipe 1.1
 You may notice that MEDAlloc 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 MEDAlloc history).

release 1.1, November 1994

MEDPlay 2.2 & 1.43, MEDRip 1.0, MEDAlloc 1.2, InstallBB 1.0, MEDWipe 1.0

release 1.0, September 1994

MEDPlay 2.1 & 1.43, MEDRip 1.0, MEDAlloc 1.2, InstallBB 1.0, MEDWipe 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 popular program which produces its own type of module is MED, which you may be familiar with as a user. A later version of MED, called OctaMED, uses similar modules. ↩

Some demos and games use MED or OctaMED modules for music 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 its running and rip out data without disturbing anything. Nevertheless, I have found this ripper software to be very effective.

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

Please note that as yet none of the programs are capable of handling the newest 'MMD2' module type which OctaMED 5 and 6 can produce. However this doesn't seem to be common yet.

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 MEDUtils' 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 MEDPlay 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. MEDPlay uses it to create file requesters and error requesters, so you should have it installed on your system before using MEDPlay.

If you use the 'Install MEDUtils' icon to install MEDPlay then this will be done automatically. Otherwise, you should first check whether you have a copy of reqtools.library by running MEDPlay 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 medplay

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

MEDPlay is a simple module player based on the
MED

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).

- You can stop or change the music being played simply by running MEDPlay again.

However, it does have a few limitations (no, really? :-), which I hope you won't find too annoying:

- ~Can't play 5-8 channel modules.
- ~Can't play MIDI instruments - only synthsounds, 8-bit samples and hybrids.
- ~Can't play MMD2 modules.

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, MEDPlay 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.

Shell usage:

For full functionality from a Shell under 1.3, MEDPlay requires the arp.library. If arp.library is not available, MEDPlay 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

MEDPlay <file> [MODNUM <n>] [LOOP]

where <file> represents the filename of the module, <n> represents a song number, and text enclosed in square brackets [] is optional. 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, MEDPlay will run through the sequence of files repeatedly until it is stopped manually. The MODNUM keyword causes MEDPlay to play just one song from a multi-module.

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

MEDPlay STOP

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

```
MEDPlay
```

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 MEDPlay in a script, please see the separate note concerning this.

Workbench usage:

There are three possible ways to start MEDPlay 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:MEDUtils/MEDPlay' (this setting would be correct if you have installed MEDUtils on a partition called 'Work'). Then, when you double-click them, MEDPlay 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 MEDPlay's icon.
- 3.~Double-click the MEDPlay 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 MEDPlay from its icon and clicking Cancel.

1.12 Using MEDPlay in Scripts

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

```
MEDPlay intro.mod
```

in a startup-sequence, e.g.

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

you will actually need to use the command

Run >NIL: <NIL: MEDPlay >NIL: <NIL: intro.mod

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

If MEDPlay 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 MEDPlay History

version 2.6, 6 November 1995

- ~MODNUM keyword implemented as in OctaMEDPlayer.

version 2.5, 14 August 1995

- ~Double-quotes are now handled consistently even in the absence of OS 2.0+ and arp.library. I felt that v2.4's approach 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 2.4, 31 July 1995

- ~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.
- ~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.
- 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.
- Arp.library is now required for most Shell use under OS 1.x
- ~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 MEDPlay in a script on a self-booting disk), reqtools.library does not need to be available.

version 2.3, May 1995

Not released.

- ~I can't remember what I changed here - see the entry for v2.4.

version 2.2, November 1994

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

version 2.1, September 1994

First public release.
 ·~Supports multi-modules.

version 2.0, April 1994

Not released.
 ·~I 'built in' the MED playroutine (4-channel songs only!).
 ·~Some OS 2.0 features used to save on code size.

version 1.43, January 1994

Released in MEDUtils r1.0 and r1.1 as MEDPlay13.

versions 1.0-1.42, 1993-1994

Not released. I spent all that time just getting the darn thing working!

1.14 medrip

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

MEDRip is a simple module ripper. This means that it is designed to identify any MED or OctaMED 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 MEDRip. Also some games intercept resets and clear their data in memory to prevent ripping. Besides which, not much game music will be stored in MED or OctaMED format. Note that I have also produced a ripper for the alternative Tracker modules as part of my ModUtils package, which may accompany this.

The effectiveness of MEDRip can be improved using MEDAlloc or MakeRipperDisk.

.

Please note that ripping is not always
legal
.

Shell usage:

To begin searching for modules, type

MEDRip

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 MEDRip 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 MEDRip History

version 1.1, August 1995

- Although MEDRip never used any OS 2.0 functions, the mode previously used to open an output window when MEDRip is started from Workbench is not supported by OS 1.x! This made the window close immediately. This has been fixed.
- I don't think MEDRip handled MMD1 modules properly in the past, but it should do now.

version 1.0, August 1994

First public release.

1.16 medalloc

Those familiar with MEDAlloc 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 MED or OctaMED 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 MEDAlloc 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 MEDAlloc 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 MEDRip 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 MEDAlloc 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 'MEDRip') 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

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

1.19 medwipe

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

MEDAlloc 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 MEDWipe 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:

MEDWipe

Workbench usage:

Double-click the MEDWipe icon.

1.20 Acknowledgements

I would like to thank Teijo Kinnunen for MED, OctaMED, and his public domain player routine (though I did have to spend many late nights hacking it into something which A68k could assemble for use with DICE's re-entrant mode).

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 MEDPlay.

//

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

1.21 How and Why to Contact Me

HOW AND WHY TO CONTACT ME

Please write or email me if you have any comments, suggestions, bug reports or programming hints to make.

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 MEDUtils and other software by me see my web page: <http://sable.ox.ac.uk/~worc0223>