

tmc

Pierre Chalamet Sync/DreamDealers

COLLABORATORS

	<i>TITLE :</i> tmc		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Pierre Chalamet Sync/DreamDealers	January 8, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	tmc	1
1.1	TMC	1
1.2	Read me first	2
1.3	Description	2
1.4	Use of TMC	3
1.5	Exit TMC	3
1.6	Depth Gadget	3
1.7	Optimize Samples	3
1.8	Create Directory	4
1.9	Slip Module	4
1.10	Global Volume	4
1.11	Convert a module	4
1.12	About	5
1.13	Change ooutput directory	5
1.14	Scroll Text area	5
1.15	Preferences file	5
1.16	TMC Errors	7
1.17	TMC Outputs files	7
1.18	TMC_Replay.s	7
1.19	TMC History	9

How to use TMC
The preferences file
Errors while converting
Outputs files
About TMC_Replay.s
History

1.2 Read me first

TMC (standing for "The Module Converter") was coded by Sync of DreamDealers.

TMC needs at least the version 3.0 of the Kickstart to work correctly but it also needs a 68020 or more to work.

TMC uses the following libraries:

```
dos.library    V39
asl.library    V39
graphics.library V39
intuition.library V39
gadtools.library V39
```

optionnal libraries are:

```
powerpacker.library V??
workbench.library V39
```

So if tou don't have the first 5 libraries don't expect TMC to work. The 2 last libraries give repectively the abilities to load powerpacked module and to drop module's icon on TMC window (AppWindow).

1.3 Description

It's a program that convert soundtracker modules to a new format so they can be played FASTER. And better, the module is sometimes really smaller and nearly unreapeable !!!

TMC also performs :

- a new patterns encoding : notes use 3 bytes instead of 4.
 - a patterns pack : TMC is able to pack blank notes. It uses only 1 byte instead of 3. But if TMC meets for example 3 blank notes on a line each after then they will also be packed on a single byte !!
 - a pattern cutting : if the end of a pattern is not used then it will be removed if it is NOT used at all
 - removes unused patterns
 - removes unused positions
 - removes unused samples
 - removes unused datas in samples (loop capabilities not to play the end of a sample if its restart is different of 0)
 - uses the system 3.0 and it's look
-

1.4 Use of TMC

When you run TMC, you must see its window appearing on the screen coz' if not it misses you a library or you don't have enough memory.

And now, Click on the gadget you want some info on:

```

+-----+
|X           NN|
|           |
|   [1] Optimize...   Create...[2] |
|   [3] Split...     Volume...[====4=] |
|           |
| [====5====]   [====6====] |
|           |
|   [====7====]   |
|           |
| +-----+ |
| |                                     | |
| |                                     | |
| |                                     | |
| |                                     | |
| +-----+ |
+-----+

```

1.5 Exit TMC

Click on this gadget to exit TMC

1.6 Depth Gadget

Euh.. Just an intuition feature !!

1.7 Optimize Samples

In case this option is OFF:

With this option, all samples will be in the song even unused samples. This option can be useful in the case you want to store several modules made with the SAME samples because you can simply change the patterns without loading the samples !! If you want to go this way, look in the 'Song.s' produced by TMC for more info.

In case this option is ON:

Unused samples will be removed.

This option is by default ON is no preferences file is found.

1.8 Create Directory

When you convert a SoundTracker module with TMC, you will get a lot of file (hum....) in your output directory which could be quite lame if you convert more than one module. So this option was made to create a NEW DIRECTORY of the same name than the module's songname (hurg !).

You can now convert a lot of module in the same directory

This option is by default OFF if no preferences file is found.

1.9 Slip Module

In case this option is OFF:

A single file (not really in fact...) is produce by TMC : it contains both song datas and samples datas. Samples and Patterns datas are still in others file.

In case this option is ON:

The first file produced is "Song.s" which contains all patterns and positions and a second file is also produced "Samples.s" which contains the samples...

This can be useful if you want to put the "Song.s" and "Samples.s" files in different section of your proggy because just the "Samples.s" file NEED to be in CHIP MEMORY (the Song.s can then be placed in FAST MEMORY).

This option is by default ON if no preferences file is found.

1.10 Global Volume

A nice feature of the version 4 of TMC is that you can set the global volume of a song. That's great if you want to do volume effect in your progs.

This option let you set the global volume AT START. If you want to change the volume while replaying then see the Replay to have a description of the use of this option.

This volume is counted in percent.

This option is by default set to 100 if no preferences file is found.

1.11 Convert a module

Click on this gadget to convert a module.

A requester appears (thanx asl !!) and you can then select a module
NOTE1: if you have the powerpacker.library you can convert powerpacked module.

NOTE2: if you have the workbench.library (your WB must be there !!) you can drop module's icon on the TMC window : it has the same effect

than clicking on 'Convert..' and choosing a file.

While converting, TMC tells you what it does in the Scroll Text area
If an error occurs then it will be written in this area like this:

```
>> prout() error <<<
```

NOTE: All file produced are written to the output directory

1.12 About

Euh... displaying credits & copyright....

1.13 Change output directory

This option let's you select a new output directory.

The output directory by default is the current directory of the CLI
or WB if no preferences file is found.

1.14 Scroll Text area

This is where TMC writes all messages it wants"

1.15 Preferences file

The preferences file "TMC.Prefs" is always an ASCII file and
is located in the S: directory.

The lines structure is:

```
<KEYWORD>parameter
```

A line can't be longer than 100 chars and can be separated
from the others with some blank lines (ONLY a Char Return
chr\$(10) is acceptable (no other shit like spaces or
tabulations..).)

If there is an error while reading the preferences file it will
be written in the Scroll Text area.

Here is all the available KeyWords in this version:

NOTE: KeyWord MUST be typed in CAPITALS !!

```
; <Comment>
```

For adding a commentary in the preferences file.

This KeyWord has NO effect. It's just for the reader...

```
SOURCE=<Path>
```

For changing the source directory. This is where your module

should be located.

DESTINATION=<Path>

For changing the output directory (When TMC has finished the conversion and wants to save the result it writes to this directory)

MARK=<Mark>

For ADDING a mark to the 'TMC build inside mark list' (arf...). It's useful for example for modules which don't have the 'M.K.' mark inside as the modules produced by the old Startrekker which write 'FLT4' instead.

The only mark in TMC is 'M.K.'

The mark MUST be a 4 letters word. You have the possibility to define binary mark by using the char '\ ' following by a 3 decimal ASCII number (I don't think this option will be usefull but nevermind !!)

If you want to use the '\ ' in the mark just type '\ '

For example : MARK=\065\066C\ will create the mark : 'ABC\'

MASKSHOW=<Mask>

For changing the show mask of the file requester.
The mask is 100 chars maximum long.

OPTSAMPLES ON

For enabling the optimization of the samples.
With this option, unused samples will be removed from the song.
This option is by default selected if no preferences file is found.

OPTSAMPLES OFF

For forbidding the optimization of the samples.
With this option, all samples will be in the song even unused samples. This option can be useful in the case you want to store several modules made with the SAME samples because you can simply change the patterns without loading the samples !! If you want to go this way, look in the 'Song.s' produced by TMC for more info.

SPLITMOD ON

For splitting the output file Song.s produce with TMC.
The first file produced is "Song.s" which contains all patterns and positions and a second file is also produced "Samples.s" which contains the samples...
This can be useful if you want to put the "Song.s" and "Samples.s" files in different section of your proggy because just the "Samples.s" file NEED to be in CHIP MEMORY (the Song.s can then be placed in FAST MEMORY).
this option is by default selected if no preferences file is found

SPLITMOD OFF

For forbidding the splitting of the Song.s file.

A single file is produce by TMC: it contains both song datas and samples datas. File is "Song.s"

VOLUME=<Decimal Number>

This let's you set the global volume.
The decimal number must be beetwen 0 and 100 inclusive
and be a 3 digits max number (??..)

SONGNAME SAVE ON

Create a directory of the same name of the module songname
in the output directory. Then all output file of TMC
are saved there.

SONGNAME SAVE OFF

Save all output file in the Output Directory.

1.16 TMC Errors

While converting or somewhere else, TMC might tells you that there was an error... I think messages are explicit so...

1.17 TMC Outputs files

TMC produce an ASCII file of your module (Song.s), which you will have to assemble with your favorite assembler
You can read the file Song.s (and maybe Samples.s) to see what kind of file is generated by TMC. The others files generated are NOT ASCII !!

TMC produces the following files in the output directory :

Song.s : it's the file that you must include
just after the replay-routine.
IT'S AN ASCII FILE SO INCLUDE IT AND
DO NOT INCBIN IT !!!
Patterns.dat : it's all the patterns of the module.
Sample##.dat : the ## designed a number in hex.
it's a sample of your module.

As the Song.s is an ASCII file, you can do what you want with it. For example, you can put the samples in chip memory and the others datas in fast memory. Song.s is dividided in 2 section the first one is the patterns etc.. and the second is the samples. Now you can understand why the converted module is hard to rip...

NOTE: you can use the Split module option to slipt the Song.s into 2 files : Song.s and Samples.s
the first one can be put in fast memory but the second on MUST be in CHIP memory (samples datas..)

1.18 TMC_Replay.s

NOTE: The TMC_Replay.s is for 68020 and more only so don't try to run it on 68000 !!!!!!!

To use the replay, just read the beginning of TMC_Replay.s (arf!)

TMC_Replay.s -only- handles the following commands :

- 0 Arpeggio
- 1 Portamento Up
- 2 Portamento Down
- 3 Tone Portamento
- 4 Vibrato
- 5 Tone Portamento + Volume Slide
- 6 Vibrato + Volume Slide
- 7 Tremolo
- 9 Sample Offset
- A Volume Slide
- B Position Jump
- C Set Volume
- D Pattern Break
- E0 Filter On/Off
- E1 FineSlide Up
- E2 FineSlide Down
- E6 Pattern Loop
- E9 Retrig Note
- EA FineVolSlide Up
- EB FineVolSlide Down
- EC Cut Note
- ED Note Delay
- EE Pattern Delay
- F Set Speed

=====> FineTune is Supported

ALL these commands work exactly as ProTracker defines them.

As you can see, it's nearly a ProTracker replay except that it doesn't handle at all PT E control commands E3/E4/E5/E7/E8 and the Invert Loop EF command.

I don't know how many raster lines it takes on a 1200 but on a my 4000/30 it use 2 rasters lines.

Something important : The replay uses the \$78.w address to initialise the samples while playing. As there is no save of \$78.w in mt_init you might have to do it yourself...

There is something great with this replay is that you can control the global volume(in percent).

There is a label in the replay called "mt_global_volume" and it is a WORD. YOU can 'move.w' there a value between 0 and 100 inclusive to set the global volume but you MUST BSR or JSR to the label called "mt_set_global_volume" to enable the volume reducer. You can change the global volume anywhere in your program.

The replay doesn't use automodified code so it must work with caches and burst mode !!!! (use it as it speed up a lot 68020 and more..)

1.19 TMC History

- version 4.2 : TMC is now loading powerpacked module. If the opening of the powerpacker.library fails then no powerpacked module can be converted.
- version 4.1 : TMC is now an AppWindow : user can drop icon on the window. If the opening of the workbench.library fails then no AppWindow will be available.
- version 4.0 : TMC is now an intuition window. It runs only on 3.0 KickStart with at least a 68020. Use of the gadgtools and 3.0 abilities.
- version 3.9 : Arf! the Window name of TMC wasn't compatible with the CON: of the 2.0 so change the window's name....
- version 3.8 : Hum... just another release of TMC... This time, it's just the replay that has been modified : it was previously using the CIA-B timer A , NOW it uses the CIA-B timer B for OS2.0 compatibility AND because it was fucking lame to reset each time I wanna use ProTracker just after replaying a zik (When using PT with a CIA interrupt...) NOW it works quite well I think !!!
- version 3.7 : Added a split module option to TMC. It's now easier to put only the samples in Chip Memory instead of the whole module.
Change SAMPLEOPT to OPTSAMPLES in preferences file.
- version 3.6 : internal release
- version 3.5 : Command E6 and EE has been added to the replay.
TMC can warn the user if a not supported command is found in the patterns.
Correct another bug in the replay : the tremolo routine was in fact a vibrato routine ... (humm.. block copy !!)
- version 3.4 : Now, if there is, on a line of a pattern, one or more blank notes each after then they will be packed on a single byte !! Now, a blank pattern have a length of 66 bytes !!
- version 3.3 : Include the commands E1/E2/E9/EA/EB/EC/ED of protracker commands : it was time to implement them !!
- version 3.2 : A blank note packer has been implemented in TMC and the replay has been modified to skip packed notes.
A blank note now uses only one byte instead of 3.
- version 3.1 : The preferences file parser has been entirely rewritten because it was quite lame.. Now it's a REAL parser which can handle infinite Key Word !!!!
- version 3.0 : TMC has been recoded at 99% and is now packing better than before by removing the end of patterns as for example after a pattern break (but only if the end of the pattern is NOT used !!)
The notes optimizer has disappeared...Maybe it will reappear one day !!
- version 2.6 : correct a bug in the replay (another one ..)
- version 2.5 : Last Change : 12 January 1993
Minors optimizations for speed purposes. The replay was losing some low notes => incrise timer value.
I also know that there is another --*BIG BUG*-- in TMC due to the note optimizer, I will fix it in version 3.0 of TMC I think it's for soon !!
- version 2.4 : Now display the size of converted module and the gain from
-

the original module. Correct a big bug in some flags which weren't clear when the conversion starts. Correct a bug in TMC with Sample Offset command (due to the note optimizer)

version 2.3 : Insert a note optimizer in TMC

version 2.2 : Optimized the Vibrato, Tremolo and Portamento routines
Correct a bug in the Pattern Break routine

version 2.1 : Added a preferences file. Yeahhh !
Correct a bug in the TMC position-list analyzer
The replay has been totally transmuted !!

version 2.0 : Now the patterns of a module are packed from 1024 Bytes to 768 Bytes. Maximum Gain is 16 Kbytes (quite bad ...)

version 1.4 : Added a finetune support and optimize a bit the replay

version 1.3 : Corrected the bugs of the Replay and of TMC

version 1.0-1.2 : let's forget thiz jokes...
