



. studio .

MIDIFILE TO CSOUND SCORE AND ORCHESTRA CONVERTER

High End Digital Audio Rendering

MIDI2CS - manual

© freeware by [Songlab](#)® 1995, 1996

at <http://www.snafu.de/~rubo/songlab/midi2cs>

. [Home](#) . [FAQ](#) . [Tutorial](#) . [Manual](#) . [Download](#) . [Csound links](#) .

Table of contents

[Idea of Midi2Cs](#)

[Composing and producing music](#)

[Installation of Midi2Cs and Csound](#)

[Start a new project](#)

[Projectfile syntax](#)

[Using sounds](#)

[Command line options](#)

[Future features](#)

Idea of Midi2Cs

Csound is the worlds leading software based audio processing system and it is for FREE! It was developed at the Media Lab of the famous MIT (Massachusetts Institute of Technology) by Barry L. Vercoe. You will find lots of links to valuable information about [Csound](#) and its use on [Songlab's Csound Link Site](#) at <http://www.snafu.de/~rubo/songlab/midi2cs/csound.html>. It might take a while to get into Csound but I am sure you will not regret learning a bit about it. Midi2Cs will be a helpful teacher.

With Midi2Cs and Csound you are able to create audio tracks of MIDI files without external samplers directly on the harddisk of your machine. You'll get rid of costs for renting expensive audio studios! The result after processing Midi2Cs's output with Csound is your own work in a professional quality.

You can get your copy of Midi2Cs for DOS, Linux, SunOS and NeXT systems on our download page at <http://www.snafu.de/~rubo/songlab/midi2cs/download.html>.

You should take a look at this online manual from time to time. Nearly every month I replace it with a newer and better manual.

For more information about Csound and music production do not hesitate to take a look at [Songlab's **Music Producer's WWW Links**](http://www.snafu.de/~rubo/songlab/musicproducerlinks.html) at <http://www.snafu.de/~rubo/songlab/musichops.html>. You will find lots of links to pages containing other useful tools to assist your work - WWW for all music production purposes!



Composing and producing music

There is a big difference between composing and producing music!

- Composing** can be done with cheapest keyboards and sequencers or just a soundcard and your computer. The result of your composition should be a midifile!

Your midifile might be a great composition, but without adequate production it won't necessarily sound like it.

- Production** is to look out for best sounds and effects, find the best equipment, studio and engineer to get the maximum out of your composition.

If you do not have the knowledge you will have to pay a good producer for doing all this work or use Csound and Midi2Cs.

You should know a little about MIDI and WAV, analog soundcards and digital I/O, sampling rates and DAT recorders to see the big chance and opportunity to be your own producer with Midi2Cs and Csound. You only need a computer with Csound installed and a program to play the created soundfiles. If you invest in a digital output to your DAT recorder none will be able to beat the quality of your production! Complex songs with nice effects may take a while calculating, on some machines you will already be able to play them in realtime without loss of quality.



Installation of Midi2Cs and Csound

You need to install Midi2Cs and Csound to render your MIDIfiles.

Csound

Download your preferred Csound version from

- [Songlab's Csound Link Site](http://www.snafu.de/~rubo/songlab/midi2cs/csound.html) at <http://www.snafu.de/~rubo/songlab/midi2cs/csound.html>.
- <ftp://ftp.maths.bath.ac.uk/pub/dream> or

Unpack it and copy it into your sound binaries directory.

Midi2Cs

Get your copy of **Midi2Cs** at

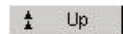
<http://www.snafu.de/~rubo/songlab/midi2cs/download.html> and unpack it.

- copy the **MIDI2CS.EXE** (DOS) or **midi2cs** (Linux, SunOS, ...) into your sound binaries directory
- copy **CS.BAT** (DOS) (a batchfile containing the line `csound -d -H -W -o %1.wav %1.orc %1.sco`) or a comparable shellscript (Linux, SunOS, ...) into your sound binaries directory.

Environment

Adding the following Csound environment variables to your `autoexec.bat` (DOS) or your shell environment (Linux, SunOS) may make your life much easier and your harddisk clearly arranged. If you'd like to know a little more about these environment settings please take a look into your Csound manual.

- SET SFDIR=C:\TMP** - directory where your songs will be created by Csound
- SET SSDIR=C:\SAMPLES** - where Csound expects the used samples



Start a new project

- Create a new directory
- type **MIDI2CS** and press enter

You will find two more files

- MIDI2CS.PRO** - project file
- MIDI2CS.CPS** - standard 'cycles per second' table for well tempered scores

Then

- load MIDI2CS.PRO** into your preferred editor and
- change 'midi2cs.mid'** into your midifile's name
- change tempo, samplerate and mono/stereo setting as you like
- set the last bar to a value like 10 (you should not calculate the whole midifile at the start of the project)
- call **MIDI2CS**

Now one orchestra file and one score file have been created with your midifiles name plus extensions `.SCO` and `.ORC` in the midifiles's directory.

- call **cs midifile's_prefix** and Csound creates the first version of your Csound score and orchestra.

You will find a `.WAV` file in your soundfile directory with the prefix of your score and orchestra.

After editing and saving new track parameters in **MIDI2CS.PRO** you only have to call again

- midi2cs**
- cs midfile's_prefix**

Your production becomes better and better depending on the quality of samples and your track settings.



Projectfile syntax

The projectfile MIDI2CS.PRO is the heart of Midi2Cs. It contains very important information about your music production (information about your midfile, stereo/mono, samplerate, used samples and many more). A valid example is written automatically by calling MIDI2CS without parameters.

The file MIDI2CS.PRO contains all options written as comments. Comments introduce themselves with a ';'. If you think that you need to use one of these options just delete the colon before you start the next compilation.

Lines that are just comments and nothing else than comments should begin with character '#' to make the file more readable.

The automatically created default projectfile

This is the automatically created projectfile with its default settings. The idea of each option is explained in the next chapter.

```
# MIDI2CS PROJECT FILE 0.94 - rubo@berlin.snafu.de
#
# This projectfile shows default options
# Change settings however you need
# (deleting semicolons, adding samplefiles, ...)

# --- PROJECT GLOBAL VALUES SECTION ---

midifile      midi2cs.mid      ; midifile (will be read)
;nameofscore  midi2cs.sco      ; score file (will be written)
;scoreheader  midi2cs.hdr     ; file to be included into scorefile (readonly)
;nameoforc    midi2cs.orc ;off ; orchestra file (will be written)

# samplerate, mono/stereo, master volume, tempo
sr      48000      ; samplerate
nchnls  1          ; 1=mono 2=stereo
givol   0.2        ; master volume
;tempo  groovesample  beats ; easy looping and syncing!
;tempo  120

# all tracks will be initialized by the following values

firstbar      0
lastbar       1000
; notrackpreselected
; transpose -24
; lesscomments
; separators

# - define default parameters and succession -
```

```

; p_pch
; p_midinote
; p_midivelocity
; p_midirelease
; p_cps midi2cs.cps
; p_maxamplitude 32000
; p_fixeddB -6

# --- TRACK SECTION ---
# define individual values for each track

miditrack      1          ;off

; sound          ; soundpath (rel. or absolute WAV-, AIF- or ORCfile)
; name dummy    ; soundname
; relvolume 1.0 ; relative volume
; instrument 1  ; number of Csound instrument

score ; all score parameters and order are optional

; firstbar 0
; lastbar 1000
; duration 1000 ; fixed duration in milliseconds
; duration sample ; fixed duration of used sample
; transpose 0 ; up or down in n halftone steps
; lowest c3 ; notes less than this will be transposed up
; highest c4 ; notes higher than this will be transposed down
; delay -0.3 ; the start parameter (p2) will be delayed

# --- parameters and succession ---
; p_pch
; p_midinote
; p_midivelocity
; p_midirelease
; p_maxamplitude 32000
; p_fixeddB -6
; p_cps midi2cs.cps

; channelsplitting instr will be instr + channel
# - if you want to use default parameters set limit with paramete
; parameters 10 ; parameters has to be the last entry
; ; in the score section (instr + start +
; ; duration + the defined parameters)

endscore
; separator
endtrack

miditrack      2          ;off

; sound          ; soundpath (rel. or absolute WAV-, AIF- or ORCfile)
; name dummy    ; soundname
; relvolume 1.0 ; relative volume
; instrument 2  ; number of Csound instrument

score ; all score parameters and order are optional

; firstbar 0
; lastbar 1000
; duration 1000 ; fixed duration in milliseconds
; duration sample ; fixed duration of used sample
; transpose 0 ; up or down in n halftone steps
; lowest c3 ; notes less than this will be transposed up
; highest c4 ; notes higher than this will be transposed down
; delay -0.3 ; the start parameter (p2) will be delayed

# --- parameters and succession ---
; p_pch

```

```

; p_midinote
; p_midivelocity
; p_midirelease
; p_maxamplitude 32000
; p_fixeddB -6
; p_cps midi2cs.cps

; channelsplitting      instr will be instr + channel
# - if you want to use default parameters set limit with paramete
; parameters 10 ; parameters has to be the last entry
; in the score section (instr + start +
; duration + the defined parameters)

endscore
; separator
endtrack

miditrack      3      ;off

; sound      ; soundpath (rel. or absolute WAV-, AIF- or ORCfile)
; name dummy ; soundname
; relvolume 1.0 ; relative volume
; instrument 3 ; number of Csound instrument

score ; all score parameters and order are optional

; firstbar 0
; lastbar 1000
; duration 1000 ; fixed duration in milliseconds
; duration sample ; fixed duration of used sample
; transpose 0 ; up or down in n halftone steps
; lowest c3 ; notes less than this will be transposed up
; highest c4 ; notes higher than this will be transposed down
; delay -0.3 ; the start parameter (p2) will be delayed

# --- parameters and succession ---
; p_pch
; p_midinote
; p_midivelocity
; p_midirelease
; p_maxamplitude 32000
; p_fixeddB -6
; p_cps midi2cs.cps

; channelsplitting      instr will be instr + channel
# - if you want to use default parameters set limit with paramete
; parameters 10 ; parameters has to be the last entry
; in the score section (instr + start +
; duration + the defined parameters)

endscore
; separator
endtrack

endofproject

```



Global options

Global options are settings that are valid for the conversion of the whole midifile. This is a list of all available global options. You do not need to edit a lot of them for your music production. The default settings are often good enough to get a fast result.

File paths

- midifile** `song.mid`
The relative or absolute path where to find your midifile.
- nameofscore** `song.sco`
The relative or absolute path of your Csound scorefile to be written.
- scoreheader** `song.hdr`
Sometimes you will need this if you are using Csound orchestra files of your own. Just put your special function tables into a separate file. Activate this option by deleting the colon in the default project file. Change the used filename `song.hdr` into yours. Now this file will be included as the top of your scorefile each time you run Midi2Cs. This saves time for editing after conversion.
- nameoforc** `song.orc`
The relative or absolute path of your Csound orchestra file to be written. It is recommended to use identical prefixes of score and orchestra files. For switching off the writing of the orcfile you should name it `dummy.orc` and ignore it.

Samplerate, mono/stereo, master volume, tempo

- sr** 48000
This option sets the samplerate your soundfile will have after creation. If you do not need the variables `kr` or `ksmps` please only use values dividable by ten.
- nchnls** 1
The number of channels your mixfile should have. A value of 1=mono, a value of 2=stereo.

Special Csound variables

- kr** 4800
Please take a look into the Csound manual.
- ksmps** 10
Please take a look into the Csound manual.

Very important stuff

- givol** 0.3
The volume output of all tracks will be multiplied by this value. Songs with many tracks should start with a value like 0.1. This makes sure that Csound does not create a distorted soundfile by cutting the high values. Then go and find the best samples and orchestras and a good mix relation for all your tracks. After that you should try to maximize the output level by raising this value.
- tempo** 120.0000
- tempo** `groove.wav` 4
The tempo of your song can be initialized by setting one of the above options.
 1. If you want to take the midifile's temposettings do not use `tempo`. Just deactivate `tempo` by using a colon or delete that line.
 2. Set the tempo you like

3. If you want to synchronize your song with an existing beat give the name of the file that contains a well cut groove and the number of beats it contains.

This option allows easy looping and syncing and therefore it is one of my favourites! Some of my old productions now have additional instruments without getting the tape from the shelf again.

All tracks will be initialized by the following values

- firstbar** 0
The first bar of your midifile you want to get converted.

- lastbar** 10
This specifies the last bar of your midifile you want to get converted.

You should know about the following:

One minute of stereo CD quality music takes about 10MB of space on harddisk and possibly also in your computers RAM. I am using a harddisk recording system that is able to read a 40MB big WAV file and play it in its own format. Simple MS-Windows soundplayers often play sound out of memory and are not able to swap it. Then the available space must be bigger as your soundfiles size!

available = your computers RAM minus 4 or more MBs for Windows

soundfilesize = nrofchannels * samplerate * nrofbars * 120/tempo

(nrofbars is lastbar minus firstbar + 1 because the lastbar is always included)

- notrackpreselected**
The default switch for conversion is on for all tracks. If you set this option no miditrack will be converted unless you specify single track options for it.
- transpose** -12
All tracks will be transposed by this value unless you specify single track tanspose options. These will be added to global options.
- lesscomments**
A lot of information about your midifile is written as comments into the specified scorefile. This option suppresses lots of comments.
- separators**
If something sounds strange in your created WAV file set the first and the last bar to the concerning range and activate this option. The result after midifile conversion by MIDI2CS and creating a WAV file with Csound is a soundfile containing the range played solo by each sample one after another.

Default parameters and succession

The intention of these options is to create default parameters for all miditracks as you normally would like to use them. These options do not work if you are

- only using AIFF and WAV files or
- automatically created sinewaves or
- if you are defining different successions of parameters for every single track

But if you are using orchestras of your own (*.ORC files) with Midi2Cs or switched 'off'

the writing of orchestra files you can activate the writing of the following parameters. The succession of parameters in the created score is identically to the succession of the following parameters in your projectfile. Take a look at option 'parameters' in the single track option list.

- p_pch**
Writes Csound pitch notation values of midinotes (midinote/12.midinote% 12).
- p_midinote**
Writes the decimal values of midinotes.
- p_midivelocity**
Writes the decimal velocityvalues of midinotes.
- p_midirelease**
Writes the decimal releasevalues of midinotes.
- p_cps** special.cps
Writes cyclespersecond values of midinotes as given in your cpsfile.
- p_maxamplitude** 32000
Set the maximum volume level. All velocity values are written as the given value divided by 128 and mulplied by midivelocity (0..127).
- p_fixeddB** -6
A decimal value that will be printed as given.



Single track options

- miditrack** 1 off
The track parameter list begins with 'miditrack' as the first word in a new line. The number '1' up to '32' selects the wanted miditrack. 'off' is the switch to mute this track - no score and orchestra will be written for this track, so that no sound can be created by Csound. Sometimes I need this when I do not want to listen to it. Just write '; off' to activate this track again. The whole miditrack ends with 'endtrack' somewhere in a new line.
- sound** filename
The relative or absolute path where to find your sample. More about this you can read in the following chapter 'using sounds'.
- name** noname
This is just to keep a little order in your project. This option is good to indentify your tracks easily in your score and orchestra files. Instead of 'noname' you can take the name of your miditrack or something else you like.
- relvolume** 1.0
The relative volume of instruments should be bigger than 1.0 for percussion and bass and less than 1.0 for other instruments.
For techno music I recommend:
global volume setting 0.15

relvolume track bassdrum 5 or 6
relvolume track snare 3 or 4
relvolume track hihat about 4
relvolume track bass about 2
and other instruments less than this.

□ **instrument** 1

If you are using your own Csound orchestra files you can choose special instrument numbers to be compatible. This options specifies the Csound instr statement. MIDI2CS 0.91 and 0.92 were not able to write orchestra files. Now it is less often used. I do not need it anymore.

□ **score**

The 'score' section ends with 'endscore'. All settings inside these markers are for manipulation of scoreparameters. If global settings exist they will be overwritten with these single track options.

■ **firstbar** 20

This specifies the first bar of your midifile you want to get converted for this track.

■ **lastbar** 25

This specifies the last bar of your midifile you want to get converted for this track.

■ **duration** 500

■ **duration** sample

The default duration of a note is the value specified by your midifile.

a) The default duration can be set to a fixed value in milliseconds.

b) The second possibility gives you the option to specify the whole length of your sample as is. Just write 'sample' instead of a value. Now Midi2Cs calculates the whole length of your sample and takes this for each note event in this track. This makes life easy! Create and change samples with soundediting tools like Cooledit and Goldwave without changing your midifile.

■ **transpose** 12

If you want to transpose your sample used in this track this is the way to do it. It will be added to global transpose values.

■ **lowest** c2

In this version of Midi2Cs you are able to use one sample for each track with automatically created orchestras. If the played range of midinotes is so big that your track begins to sound silly just specify this. Your note will be transposed up in octaves until it is higher than the set note.

■ **highest** c4

Your note will be transposed down in octaves until it is lower than the set note.

■ **delay** -0.3

If you want to get a running groove you should try this. Most samples do not have a strong attack and so it is a good idea to (minus) delay the

related samples a little to make them sound earlier. Then you have the result that there will be a better synchronisation between points of highest intensity. That hits man !!! (The delay option affects Csound parameter p2.)

■ **p_pch, p_midinote, p_midivelocity, p_midirelease, p_maxamplitude, p_fixeddB, p_cps**

All these parameters are identical to global options but only valid for this track.

■ **channelsplitting**

If this option is set, Midi2Cs adds the midichannel to your instruments number. I use it to demix tracks with more than one instrument. Those tracks may be drumtracks or tracks with different instruments. Do not forget to set option instrument to a value somewhere above 16 to avoid duplicate specified instrument numbers. Orchestras are not written automatically. You need to create an ORC file that contains the wanted instrument specifications.

■ **parameters 7**

If your orchestras always seem to have the same parameters and succession and just differ in the number of parameters you should think about using global default parameters. This option exists to set the limit of parameters that will be printed into your scorefile. 'parameters' has to be the last entry in the score section. Do not forget to add the three Csound default parameters 'instr', 'start' and 'duration' to your parameter count.

□ **endscore**

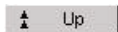
Do not forget to mark the end of your score section 'score' with 'endscore'.

□ **separator**

This one will split your soundfile in two parts - the tracks you created before including this and the tracks you will create after this one. Read also the global option 'separators'.

□ **endtrack**

This one marks the end of your track section. Now Midi2Cs is looking for the end of the projectfile or a new miditrack description.



Example

This example shows how your projectfile could look like after some editing.

Hottest Tip:

If you are using AIFF files with notes in the beginning or at the end of the filename's prefix the created Csound tables are tuned for sampling already!

```
midifile          schauf.mid
nameofscore       midi2cs.sco      ; score file (will be written)
;scoreheader     midi2cs.hdr      ; file to be included into scorefile (readonly)
nameoforc        midi2cs.orc      ; orchestra file (will be written)
```

```
sr      48000    ; samplerate
nchnls  2        ; 1=mono 2=stereo
givol   0.3      ; master volume of 0.3 seems to be good for many songs
;tempo 95.4534
tempo d:\studio\samples\rec\bwhite\t7t2.wav 4    ; easy looping!

# all tracks will be initialized by the following values

firstbar 35
lastbar  57
notrackpreselected

# values for single miditracks

miditrack 1
  name Reaggae
  sound dev/ek44/jazzorgn/d3_m.aif           ; relative path to SSDIR
  ;sound div/welle/d#3_s48k.aif           ; this is just another comment!
  score
    transpose 12
    delay -0.04
  endscore
endtrack

miditrack 2      off
  name Strings
  sound orcs/strings.orc
  score
    p_maxamplitude 32500
  endscore
endtrack

miditrack 3
  name Clavinet
  ;sound rec/royorbis/d4_har.aif
  sound rec/canufeel/f2acid10.aif
endtrack

miditrack 4
  name Snare
  ;sound rec/bwhite/sn3_s48k.wav
  sound rec/gift/sn1_s48k.wav
  relvolume 1.5
  score
    duration sample
  endscore
endtrack

miditrack 8
  name Snare2
  sound dev/dr110/sn1_s48k.wav
  relvolume 0.5
  score
    duration sample
  endscore
endtrack

miditrack 10
  name Sequence
  sound own/spieluhr/d4_s48k.aif ; relative path of a sample,
endtrack

miditrack 11
  name Bassline
  relvolume 2.5
  ;sound own/bassysa/c1_m48k.aif ; relative path of a sample,
  sound dev/ek44/kickbas2/a0_m.aif ; relative path of a sample,
```

```

        score
            transpose 12
        endscore
    endtrack

miditrack      12
    ;sound D:/studio/samples/dev/tr707/hh1_s48k.wav
    sound D:/studio/samples/own/fatima/hh1_m48k.wav
    name Hihat
    relvolume 2
    score
        duration sample
    endscore
endtrack

miditrack      13
    ; sound D:/studio/samples/dev/tr707/oh1_s48k.wav
    sound orcs/oh1_48k.orc
    name OpenHihat
    relvolume 0.3
    score
        ;duration          sample
        delay -0.2
    endscore
endtrack

miditrack      14
    name BassDrum
    sound D:/studio/samples/rec/sly&rob/8/bd1_s48k.wav
    score
        duration          sample
    endscore
endtrack

miditrack      15
    name Groove
    ;sound d:\studio\samples\rec\bwhite\t7_s48k.wav
    sound d:\studio\samples\rec\bwhite\t7t2.wav
    score
        duration 4250
        delay 0.1
    endscore
endtrack

endofproject

```



Using sounds

There are some ways of using sound in Midi2Cs.

You can use

- WAV files (samples)
- AIFF files (samples)
- sinewaves (automatically created Csound orchestras)
- orchestras (manually created Csound orchestras)

WAV files

WAV files are easily produceable on a PC. Nearly every audio program uses them and a lot of archives exist where you are able to ftp them. You can use them with Csound for all tracks where there is no need to change the pitch (e.g. drum tracks).

AIFF-files

If you want to play melodies with the current Csound version for the PC you need to use AIFF files.

You do not need to understand the specification of AIFF that can be found on SGI's FTP server <ftp://ftp.sgi.com/sgi/aiff-c.9.26.91.ps.Z>.

WAV files can be easily converted to AIFF files by using the program **SOX** (SOund eXchange). You will find it at its location <http://www.spies.com/Sox/>.

Sox has a lot of build in features, but I think you will not need them. Just call e.g.

```
sox file.wav file.aif
```

and your AIFF file has been created.

Sinewaves

A simple Csound sinewave orchestra will be created automatically if you are not defining any sound.

Orchestras

You want to use existing Csound orchestras with Midi2Cs ?
Just do the following:

- Copy a single orchestra sound description into a separate file with extension **.ORC**
- Add **;ORC** to be the first 4 letters of this new file
- Delete the **instr** line or specify it as a comment. This line will be automatically added by Midi2Cs.
- Delete the **out** and **outs** line or specify it as a comment. This line will be automatically added by Midi2Cs.
- Take **aright** and **aleft** as output variables.
- Add necessary functions and tables to the "midi2cs.hdr" file

```
;ORC ---- orchestra.orc ----

;
; your orchestra stuff
;

      aright = ... ; to be specified
      aleft  = ... ; to be specified
```

Once you have done this you can use your orchestra with every track by just specifying it as

```
sound filename.orc
```

in the track section of the MIDI2CS.PRO file.



Command line options

If still someone uses command line options please [send a mail](#). Otherwise we'll drop command line options with our next version!

Maybe using a simple command line option may fit your purpose.

All '#' characters should be replaced by numbers.

Do not type blanks between option specifiers and values (numbers and filenames).

Calling **midi2cs** with option

- h (help)**
- ? (help)**
prints a short option overview to the screen
- k (known bugs)**
lists known bugs and additional features

(You can use and mix as many of the following options as your commandline fits.)

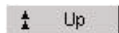
- a (append)**
appends the output to an existing scorefile
- b# (begin)**
specifies the first bar of your midifile to begin with
- d#,#[,#[,#[,#[]... (duration)**
specifies the duration of your instrument by typing e.g. -d3,2500,5,350. This means that all notes of track 3 will have the length 2.5 seconds and all notes of track 5 will have a duration of 0.35 seconds
- e# (end)**
specifies the last track of your midifile to be converted, e.g. -e30
- f"filename" (file)**
includes the specified file to the top of your scorefile, e.g. by typing -fdummy.hdr
- i#,#[,#[,#[,#[]... (instrument)**
assigns each track a specific instrument number, e.g. -i4,23,8,9 assigns instrument number 23 to track 4 and instrument number 9 to track 8
- l (less)**
suppresses some unnecessary comments in your scorefile
- o"filename" (out)**
assigns the name of the scorefile, e.g. -odummy.sco creates score file dummy.sco
- p#,#[,#[,#[,#[]... (parameters)**
Drumtracks often just need the 3 score parameters instr, start and duration. This option suppresses the writing of unnecessary parameters. E.g. -p7,4,3,3 causes

that track 7 will have 4 of the default parameters and track 3 just the minimum of 3

- r"filename"(read)**
reads the projectfile. Syntax errors will be printed out to the screen. See the syntax of this powerful option in chapter projectfile
- s#[,#]... (select)**
selects only some tracks to be written, e.g. -s3,6,12 tells Midi2Cs to extract tracks 3, 6 and 12 from your midifile to be written
- t#[,#,#]... (transpose)**
transposes tracks, e.g. -t3,5,7,-7 transposes all notes of track 3 five halftones and all notes of track 7 seven halftones down

The following is an example of how to use the described command line options in a .BAT file. A good projectfile however is preferable.

```
rem 6 Bass
midi2cs -b34 -e51 -l -s6 -i6,20 -p6,4 -ffatima.hdr fatima.mid
rem 7 Bassdrum
midi2cs -a -b34 -e51 -l -s7 -d7,800 -i7,1 -p7,3 fatima.mid
rem 8 Snare
midi2cs -a -b34 -e51 -l -s8 -d8,1000 -i8,2 -p8,3 fatima.mid
rem 9 Hihat
midi2cs -a -b34 -e51 -l -s9 -d9,190 -i9,3 -p9,3 fatima.mid
rem 1 Juno
midi2cs -a -b34 -e51 -l -s1 -i1,21 -p1,4 fatima.mid
rem 3 Strings
midi2cs -a -b34 -e51 -l -s3 -i3,21 -p3,4 fatima.mid
rem 5 Sequence
midi2cs -a -b34 -e51 -l -s5 -i5,21 -p5,4 fatima.mid
```



Future features

The aim of Midi2Cs is to produce music of highest quality as easily and as cheaply as possible. There are lots of ideas that will be implemented in the near future. Please tell me which option seems to be the most important for you to become a user of Midi2Cs.

The following list shows my current plans for the future. The list, its succession and the speed of implementing functions may change depending on your interests.

- General MIDI support
- Automatic creation of projectfiles relating to the real miditracks
- Better include of standard (non sample) Csound orchestras
- Adding effects like panning, delay, reverb
- Windows Desktop
- Resampling "sound" samples depending on samplerate value sr

Thanks goes to all who have provided feedback, and have helped to make this a better product. The responses received have had a direct impact on the evolution and improvement of Midi2CS.

Send comments or suggestions to my E-Mail address rubo@berlin.snafu.de.

SongLab
"Midi2Cs"
Bonhoefferufer 13
D-10589 Berlin

▲ Up

Mail to RuBo@ Berlin. Snafu. De

Next Page ►►