Helpfile for m2buzz (GUI version 3.62; most help topics still refer to 3.0):

Contents

Options (the single tabs)

Glossary

Introduction

The program BUZZ by Oskari Tammelin is a very promising tracker of the 4th (?) generation, if not more: software synthesizer, effect engine...
Despite it's alpha-state a lot of people already have started composing songs with it.
Hey, and it rocks: if you've never tried it, just get your hands on it and start playing around and come back here only if you need to ;)

For me it's great potential comes (additional to the modular machine design) from the as well sequential as pattern-oriented approach, kind of a mix of a sequencer and a tracker. While I find MIDI sequencers best adapted to an everchanging flow of music, a sequence that is sometimes hard to manage, trackers used without care often produce pure periodicity, which can get boring very fast. In BUZZ you can start with simple patterns and evolve your tune as you want it to: let it stay periodic or abandon the rhythm completely, come back to where you started or finish your song far away.

The bad olds is: in the present state (buzz 1.0) there is only rudimentary MIDI-input, and no MIDI-recording at all. I also couldn't get pattern information into BUZZ by other means than by keyboard input in BUZZ (not via clipboard for instance). So I decided to do some quick coding of a converter from MOD-files to BUZZ-songs, based on the existing source of a MOD-file dissassembler MODDIS by David Lai. After several days mod2buzz v1 was born, and I could at least record some MIDI with Fasttracker II, export it as MOD-file, convert it with my program and finally load it into BUZZ. But Fasttracker's MIDI-handling isn't very exciting (although it was in former times ;) and none of the existing MIDI->MOD converters could convince me. This (and a friendly mail of Oskari Tammelin :) let me go further and finally we are here:

m2buzz v3 Features

Return to Contents
Next Topic: m2buzz Features

Features

m2BUZZ (version **3**) can now handle MOD files, XM files and MIDI files and convert them to BUZZ songs. I mainly concentrated on the pattern information, especially all note events, and timing. But I also implemented some effects and other things, so a feature list probably would be larger than the Missing Features. You can find a detailed progress report in the History section of this helpfile.

The highlights of version 3.x are:

-conversion of Fasttracker XI instruments into an almost empty BUZZ song with only the master machine and one instrument, which may be useful for the equal new

-merging and **transforming** feature for existing **BUZZ songs**.

-conversion of uncompressed Impulse Tracker 2.xx files (**IT**) (this is probably no IT2.14+) by means of the it2xm converter by Andy Voss and Screamtracker modules (**S3M**) with MOD2XM by Arnaud Hasenfratz (get these converters from MAZ).

-support for other (generator) machines than just the tracker only

-basic wave and ES-9 data ripping, wave tuning to non-standard values

Since version 2.1 converted **modules and XM** files contain all samples and wavetable data in the song and don't rely on the BUZZ tracker sample loader anymore. The news for MOD / XM conversion compared to version 2.1 are

-the **splitting** of every instrument to a different machine (every track since v2.0)

-**removing** of unused and duplicate patterns

-conversion of volume information and **envelopes**.

- effect conversion for Matilde Tracker machine

Finally the **MIDI** conversion has been improved significantly, an additional converter for **GUS-patches** easily creates a complete patchset for this purpose, but you may also use the special support for my **vMidiOut** machine (transfer of channel and patch information additional to notes and velocity).

I guess most things work satisfying now, but some -especially optical- things aren't ironed out yet, hence the constant beta state. Because of the announced format incompatibilities between the current BUZZ and the upcoming BUZZ2 the future of m2buzz is kind of uncertain, so I don't want to put a lot of efforts in unneeded things and the beta state probably remains.

Previous Topic: Introduction
Next Topic: Merging

Merge/Transform

Starting with version 3 a new tool is integrated with m2buzz - the BUZZ song merger/tranformer b2buzz. It generally allows to merge two BUZZ songs into one, giving you the possibility to access the patterns and instruments of another song, which currently is kind of complicated if not impossible with BUZZ 1.x. Since version 3.5 you can even concatenate them sequentially.

Main purposes are

- adding of a single voice (after MIDI conversion: this allows you to play a MIDI voice along with a BUZZ song, record it and finally merge it into the original song),
- use of existing patterns (for remixes and the like, just no copy and paste anymore ;)
- import multi-sampled instruments (after XI conversion: you can edit the parameters like pitch and loop points with Fasttracker and get them right into BUZZ).
- adding of single machines with a predefined setup (no need to adjust the parameters everytime the same way)
- complete merging of several sequences (but take care: the 112 pattern limit of BUZZ 1 remains)

The BUZZ-fileformat doesn't contain some machine dependent information, which BUZZ gathers by using the corresponding DLLs. m2buzz reads this partially from a supplied file machines.ini. If you encounter an unknown machine or play a BUZZ song a file machine2.ini of similar format will be written with information gathered from the machine DLL's in your BUZZ gear directory. If present this file will also be, but machines.ini has still to be present.

Additional notes:
In the meantime I've added the blowup feature: adding empty lines to existing patterns while raising the BPM rate of the BUZZ song, very useful for MIDI conversion+merging. The possibility to repeat the whole sequence of the song may overcome the missing copy&paste in the sequence editor and finally the "cloning of a machine with some mutations" (somebody have a better name?) stands for itself: now the merging isn't the main purpose here anymore: you can transpose a machine, create another generator with the same or transposed notes, replace samples of a tracker machine... You also can change samplefrequencies to values beyond those allowed as input in BUZZ.

Previous Topic: m2buzz Features
Next Topic: Basic Options

Basic Options

If all you want is to convert a MOD or XM file, this tab with the basic options should be all you have to care about.

Input file

Choose the input file with the Browse button, supported types are:

MOD**:** 4/6/8 track modules
XM**:** Fasttracker II extended modules
MIDI **:**

With the *internal* MIDI-parser you currently only get note event conversion, but every midi-track is routed to one buzz-machine, if you want to have only one buzz-machine, convert your midi file to midi format 0 first. Also supported is the external MIDI->XM converter EasyXMid.

XI: Fasttracker II extended instruments

They are converted to an almost empty song with only the master machine and one instrument, this is useful for adding an instrument to an existing song with the merging option or simply to start a new song with a cool drumset ;)

IT: Impulse Tracker module

These are only supported via the external it2xm converter by Andy Voss (get it from MAZ), it is started and the resulting XM file will be processed. Save your IT's as version 2.xx ones in Impulse Tracker , not as v2.14+ compressed modules. The later may give unpredicted results (but generally an error message). The XM file may overwrite an existing one without warning and won't be deleted after conversion. This conversion is of course additionally limited by the limits of it2xm (no virtual channels stuff,...), but the results seem to be o.k. (I've done almost no testing).

S3M, BMW, BMX:

Only some logging information is given, no conversion is done (yet ;), but BMX files are supported as input in the merging tab.

Play: Only the MOD formats can be played (-> no MIDI), the player uses the BASS lib by Ian Luck. Press the button again to stop playback.

Output file

You probably guess what this does ;) If the wave options are changed, I try to determine the extension BMX or BMW automatically, if you choose **Make BMW** for a MOD-file, a waveless BMX file with probably the wrong BMW-extension will be created, take care of this.
The resulting songlength is truncated to 112 patterns (BUZZ songs) and a check for the maximal polyphony of every machine is done.

Splitting to machines

The split options decide about the number of machines you'll have in your converted song for MOD and XM conversion. This doesn't effect the internal MIDI conversion, you'll have to convert the input format of your MIDI-file to format 0 or 1 before to get different results.

single target machine/ no split:
You get one single BUZZ tracker machine. I've found its maximal polyphony to be 18, you'll get an error if you try to use more tracks. If you use other machines as conversion target the polyphony generally is lower, you can change it in machines.ini, but I think the values there should be correct already.

assign machines/split by track:
You'll have one machine for every track. This is the recommended option.

assign machines/split by instrument:
Every used original instrument gets its own machine, the number in the machine name is equal the instrument number for better recognition. The number of tracks for one instrument is equal the number of original tracks, where this instrument occures, no further track reduction is tried (that is if you play sample 1 in pattern 1 on track 1 and in pattern 2 on track 2 you'll get two tracks for the machine representing the first sample).
If in the original track an instrument number changes, the converter places a note-off command in the machine with the old instrument, while proceeding with the new one. If a pattern starts with notes or effects without a given instrument number, it is tried to determine the active instrument according to the last explicitly set instrument in the pattern sequence of the song before the actual pattern. This may be wrong, if the previous pattern ends with a pattern break, but additional data follows, or if there are more than one possible precedessors of the pattern. If the pattern is the first in the song, the last instrument of the song in the current track is used (this mostly concerns noteoffs for loop preparation).
I guess this option now works as intended, but you get more machines and tracks than with the previous option, so better you only use this if you need/want to.

Instruments/Waves

You probably want to leave this untouched in the "**BMX with Waves**" position for simple module conversion. If you want to choose "**Make BMW**" instead, have a look at the XI-directory in the MIDI-tab , but the default should be o.k. The alternate machines can give you a lot of fun, but also some problems with polyphony.
For conversion to the vMidiOut machine you can already choose here a MIDI device to avoid having to change it for every machine in BUZZ. If you convert a MIDI file with the MidiOut machine as target, additionally to the notes and velocities the patch and channel information is retained, so all voices should be converted correctly (without controllers and tempo changes). This works best for type 1 MIDI files and you have to use the following options in the MIDI-tab: "**use programs**", "**drums follow pitch**" and "**pitch correction: 0,0**"
New in v3.6: Look for info regarding the generator list

Play: The BUZZ player uses the buzzplay lib by Jeskola. It refuses to play some songs and casually crashes the whole programm or computer, so take care. Press the button again to stop playback.

The big buttons at the bottom of the dialog do the main work ;)

Convert

launches the conversion with the current options from the blue tabs,

Transform

launches a transformation or merging task the current options from the red tabs,

Save

saves the current options to the ini-file, except *tracklimit;input file,output file, logfile.* The ini-file is used by the Windows and the DOS executables.

Quit

does nothing else than quit the program without saving anything and

Help finally does show you these fine help lines ;)

Previous Topic: Features
Next Topic: Basic Options: Instruments/Waves

Instrument conversion (Basics tab)

use Tracker machine:
BMX with Waves
This is the recommended option, all waves are written into the BUZZ song, finetune and transpose of the samples should (hopefully) be correct.
For MIDI-files you need to supply instruments in the XI-directory.
If a temporary file m2buzz.tmp still exists after the conversion, you can safely delete it.

make BMW
The wavetable information is included in the song, but no wavedata, the waves are (or need to be, look for the info below about the single filetypes) stored as XI-instruments in the XI-directory and loaded from there while loading the song with BUZZ. Since the complete path to the instruments is included with the song you can't move these instruments to another place or distribute the song this way.

Often the loop settings (flag,start and end) are set wrong for multilayered instruments (probably some information stored in the BMW-file is overridden by the BUZZ XI-loading routine). This also occures for files saved with BUZZ :(

for XM:
All instruments are ripped from the song and stored into the XI-directory**.** Take care: there's no test for overwriting done and beware of the too many instruments in one wavetable-directory error of Buzz a15("Wave Tree Bug"?) if you include this directory in the BUZZ wavetable section.

for MIDI:
You have to supply XI-instruments to make this work, there is only a minimal set for the **use channel** option in the package (MIDI-options).

for MOD:
This option will be ignored and a BMX-file without waves will be created (take care of the bmx-extension yourself, or simply use the next option).

no Waves
Neither waves nor wavetable information is included, you can load a **MOD**-file with the Jeskola Tracker machine, but you won't have the correct samplerates

use other machines:
All notes are put into the corresponding machine instead of the tracker, no other effects are processed; you have to adjust the parameters of the machines yourself (or change them in machines.ini). This option often yields very experimental results but also a lot of fun and it's especially useful for MIDI -files without drums. The results for MOD and XM files depend on the tuning of the used samples, drums generally give unwanted noise.
To selectively use other generators, you could convert your song using the tracker and than clone a single machine with the transformation options.

There may arise problems with the maximal polyphonie of the machines: While you can use up to 16 channels with the tracker, this may be different for other machines, the organ for example really only allows 8 channels, the bass machine only 4.
This option relies on the correct info in machines.ini
For the ES-9 you need additionally a file es-9.dat containing the data you can change in BUZZ   with this nice edit screen (which often misbehaves ;). I have put one in the package, but you can easily rip the data from any existing song with such a machine by choosing the **Rip ES-9** option in the Machine Transformation tab. Take care with this, as this machine sometimes misbehaves as its editor: depending on the data the sound output may freeze, I've got some songs which I have to load several times until I really can listen to them again, sometimes I even have to restart BUZZ :(

New in v3.6: Look for info regarding the generator list

Previous Topic: Basic Options
Next Topic: Extended Options

Extended Options

These options you'll probably never touch, maybe with exception of transposing and window coordinates.

Song conversion

Transpose
This allows you to transpose all notes in the song (half tone steps, 12 would be an octave).

extern MOD conversion
The converter mod2xm I use for S3M also allows (just look at the name ;) to do MOD files, so I'm offering the option to use it instead of my own intern routines.
I didn't do a lot of testing or comparing the results, but an advantage of doing this may be the possibility to rip the XI instruments of the resulting XM with m2buzz and the option to create a BMW file this way.

Limit tracks
Use this if you don't want to convert all tracks, especially for XM's created by EasyXMid.

start numbering with zero
Default is to number the machines beginning with one, if you split by tracks and want to have machine numbers equal to the tracknumbers in FT2, you should check this option.

Remove duplicate patterns
This option is checked as default and removes most of the lot of pattern duplicates, which get created if you convert modules with any of the split options. This helps you to work further with the song in BUZZ, but makes it more diffult to find the original pattern in the module, so you can switch it off.
Other advantages of this option are a reduced size of the converted file and the virtually raised pattern limit, because the pattern limit of 112 does of course only affect the patterns per machine in the resulting BUZZ song.
Currently not removed are duplicates of patterns which are created by patternbreaks with an offset into the next pattern in the input file.

Shorten Song / Length - 1
Some XM's not created with Fasttracker II have a wrong songlength or wrong patterns in their songlist. The converter replaces invalid patterns automatically with the last valid, but sometimes you may want to simply shorten the song by one to get the correct results.

Effect conversion **(not a lot here ;)**

Use logarithmic volumes
This is almost completely untested and shall improve the volume handling. Currently

only the set volume command and the volume column of XM's are concerned, no slides yet.

Wave conversion

Add zero loops

Before BUZZ 1.0 the tracker machine introduced a lot of unwanted clicks, even with correctly zero-terminated sample-data. Some of them could be removed by adding a loop over the last sample for a originally non-looped sample, which is done with this option on. But this probably is obsolete now, as the recently released fixed tracker machine finishes all the trouble :)

Use Envelopes

Default is to not convert XI/XM envelopes, because this is a recent feature and not too much tested. But the conversion should be o.k. now, the envelope length of tracker machines is set to 324 frames as in FT2. Still missing are a correct release phase (especially the fadeout of FT2) and a nice implementation of the ADSR stuff.

Process control

You can log a lot of file and conversion information to a file; probably not very useful for anybody else than me, but if you check "**Verbose**" you even learn the names of the effects ;) These options are basically remains of the DOS version, in this 3beta you best leave them unchecked, the kill button also won't work. You can change the window coordinates, but not all values for these work correct yet.

If you want to start BUZZ from the m2bgui with the **Start BUZZ** button, you currently have to save an ini-file with **Save** and manually edit the entry **buzz_path** to the directory where your BUZZ executable is located (inclusive a final backslash!) and than restart m2buzz. In later versions I may read this directly from the registry.

Previous Topic: Basic Options: Instruments/Waves
Next Topic: MIDI Options

MIDI options

Midi resolution(TPB...Ticks per Beat/ BPM res)

The former midi resolution setting for the internal conversion is now replaced by two values: the BPM resolution determines the ratio of the resulting BPM setting in the BUZZ song to the initial BPM rate of the MIDI-file and the TPB value gives the BUZZ ticks-per-beat value for the MIDI file. This approach seems more reasonable; the complete resolution is **BPM-res*TPB/4**, so instead of a former midi resolution of 6.0 choose a BPM-res of 1.0 and a TPB value of 24.

lines/pattern

The default of 64 lines in a pattern generally should be allright, but in special cases, for instance to avoid the 112 pattern limit, you may want to change/ increase this number. You are responsible to use reasonable values.

scale time

With this option you can get a more accurate timing, if you need this, for instance in case you want to merge a new voice into an existing BUZZ song. The default is 1.0, if you increase this value, the song will slow down, although the BPM rate stays the same, and the number of rows for the whole song will grow. The timing change is equivalent to a by the scaling factor reduced BPM rate in BUZZ, but in this case the pattern data would not change. To change the BPM rate only is easy to achieve in BUZZ itself, or by corresponding changes of the MIDI resolution above and the scaling factor: if you double the resolution and divide the scaling by two, you'll get the same patterns with double tempo.

shift time

This option allows you to synchronize the converted sequence by shifting it in time. A number of 1.0 is equivalent to one row in BUZZ, with negative values the start of the sequence will be truncated. This is similar to pressing insert or delete in a step view of 1 in the sequence editor, but allows fractions of a row.

start with first note

Instead of using the whole MIDI time of the file, you can choose to start your patterns with the first note event. This skips the empty time in the beginning of the song and may also useful for synchronising an additional voice to an existing song.

use note delay

This was intended to get a more accurate timing by using a tick division of 15 and adress times of 1/15 row in BUZZ. But principally this doesn't cure the problem of short notes and unfortunately note delays for note offs currently seem not to be supported in BUZZ and I didn't try this yet with note-cuts. So you get a better resolution for the beginning of a note, but the length often is worse, so it's recommended to leave this unchecked.

use Programs

You get one wavetable instrument for every used patch in the MIDI-file, the complete drumset (drumchannel 10 is assumed) will be the first intrument.

You must supply a complete patchset of XI-instruments in the XI-directory named **gmp_01.xi**...**gmp_80.xi** (hex numbers: 09,0a,0b,0c,0d,0e,0f,10,11 and so on) for 128 melodic instruments (General MIDI assumed they are Accoustic Piano ... Gunshot). Normally a set of drum files in the range of **gmp_81.xi...gmp_ff.xi** is used, as created by gus2xi (see below), but with checked **multi-sampled drumset** option a file **gmp_00.xi** containing the whole drumset is used instead (for this the otherwise empty first place in the wavetable entries   is reserved). With such a drumset drums are nicer to manage but as disadvantage the resulting songs will always be very large.

You can make the instruments by converting any patchset for the GUS or AWE (AWE patchset should be without ROM-samples) with Awave or another sample/instrument converter or you can load GUS-patches with Fasttracker II and save them as XI-instrument . (Yes, that's some work to do, but my patchset is about 10MB, alone 1.5 MB for the drumset, too large to send it to anybody, I converted it from the guspro2 patchset).
Now you can use my **gus2xi** converter to create a complete patchset from a set of GUS-patches. Just copy your ultrasnd.ini to your XI-directory and type "gus2xi ultrasnd.ini" (take care, old XI-patches will be overwritten without warning).

You should also check this option (use programs) for converting MIDI files to the vMidiOut machine.

use Channels
Only one instrument is used for every MIDI-channel, the instrument names have to be **gmc_01.xi** ...**gmc_10.xi** (hex numbers again, drumset generally would be gmc_0a.xi), a rudimentary set is supplied (only one "General Drum" sample as gmc_0a.xi and one "General Melody" sample for the rest). This is really only intended to get you going with MIDI and the supplied instruments, don't worry if the results sound very bad. Exchange the patches, if you just want to experiment a bit, but for serious use you should get or create a complete patchset and use the first option.

extern midi converter
Use the extern Midi->XM converter **EasyXMid** instead of the internal routines.

multi-sample drumset
Use one single XI file for all drums, look above (**use Programs**) for details.

The next options deal especially with patches converted by **gus2xi** and apply only if the option **multi-sampled drumset** above isn't checked. The inconveniences are caused trying to have as correctly converted XI-files from the GUS-patches as also finally a correct sounding converted MIDI-file. I'll look for a better solution in the future...

remove loop-flag from drums

Some of my GUS drum patches have a loop-flag set, although this sounds terrible after conversion. Until I find a correct solution, I leave the loop in the XI-file and you can remove it here.

drums follow pitch
Should not be checked with **gus2xi** patches, so drum XI's are always played at C-4 +transpose + drum-correction.
You have to check this option for converting MIDI files to the vMidiOut machine, so the different notes played at the drum channel 0x0a really trigger different drums and not only the one for C-4.

pitch/drum correction
The melodic patches have the correct pitch in FT2, but are too high in BUZZ. As I don't want to change XM and XI conversion (as probably the mostly used subjects of m2buzz) only MIDI-notes have to be transposed down, but without the drums. Here you can choose additional values for transposing melodic and drum patches. For correction of the **gus2xi**-patches you should currently (gus2xi v0.5, m2buzz v3d) use -12 and 0 here.
For conversion to the vMidiOut machine use 0 for both.

XI Directory
Here is looked for the instruments to use for MIDI conversion, create one new directory for this single purpose and set it's path here (the Browser doesn't work perfect :(just type it in, you need to do it only once :) or just use the default XI-subdirectory of the working-dir. This directory is also used to store the ripped XI's during XM-conversion with checked "**Make BMW**" option.

Previous Topic: Extended Options
Next Topic: Transformation Options

Transformation options

Read first about the merging feature, if you don't know what to do here. The feature has expanded very much, so merging of files only is a part of what you can do here.

Inputfile I
Here you choose the first song, which will be transfered unchanged into the merged song or may be transformed with the following options, if a complete and correct machines.ini exists. This is also necessary for the machine transformation options in the next tab. Generally only one of the options: merging, blow_up, repeat or cloning should be used at once.

scale machine positions into upper-left
To avoid the graphical mess in the machines view after merging, all machine positions of the first song will be scaled into the upper left part of the screen.

add machine prefix "A "
Merging two songs may result in unwanted behaviour of BUZZ, if you have machine with the same name in both songs. If you put different prefixes in front of all machine names in both songs, you can be sure to avoid this.

blowup
This feature allows to increase the tempo of the first song and equally all pattern sizes by an integer factor **blow_up** by adding **blow_up**-1 lines between every existing pattern line (but both, a value of 0 or 1, don't change the song). A correct machines.ini containing all machines of the song is needed. The data for an empty line for generators is taken from there, for effects the existing lines are simply multiplied. In patterns for the master machine existing BPM changes are adapted, also the sequence section. Problems with this feature arise from effects or machines which use fractions of rows as explicit time values. This includes a lot of tracker effects and for instance the gapper. This feature is intended to improve merging MIDI voices into existing songs.

repeat sequence
BUZZ currently has no cut&paste in the sequence editor. To get you started more easily with repeating parts of the song, the whole sequence of the song may be repeated. You can delete superfluous stuff and shift the sequences in BUZZ. Take care to set the song end marker correctly before you use this option.

Inputfile II
If you want to merge two songs, here you chose the second. If you only want to transform the first song, choose **none** here. If you check **use conversion output** the name in the outputfile field of the basic options tab will be used as second input for merging and the next two options will be checked automatically.

scale machine positions into lower-right
As for the first file all machine positions of the second song will be scaled into the lower right part of the screen. Read above for details.

add machine prefix "B "
Merging two songs may result in unwanted behaviour of BUZZ, if you have machine with the same name in both songs. If you put different prefixes in front of all machine names in both songs, you can be sure to avoid this.

shift tracker waves
If the second song contains tracker machines, while the first contains wavetable entries, the tracker will play the wrong instruments, if their numbers aren't increased by the number of instruments in the first file. This should also work for other machines using the wavetable like vGraphity or Arguelles Pro2.

sequence merge (with master)
This is the new (with version 3.5) option to concatenate two songs completely. Additionally to normal merging the following things will be done:

- shifting of all sequences of the second in time (take care to set the song end of the first song correctly!); the start of the second song will be set to a multiple of 64 for better overview
- keeping all master patterns of the second song, using prefix "A " for patterns of the first, prefix "B " for patterns of the second song (absolutely necessary, BUZZ sorts patterns alphabetically after loading, without reordering the assigned patterns in the sequences)
- keeping master sequences and connections of second song
- create two additional one-line patterns "Setup A" and "Setup B" for the master with the default values saved for both songs and put them in an additional master sequence

It is recommended to check additionally the "**shift tracker waves**" and both "**add machine prefix ...**" options. Take care with the 112 pattern limit for one machine of BUZZ songs, which also applies to the master.

merge after conversion
With this option checked the merger/transformer is automatically launched after conversion. This probably only is useful to merge the converted file with an existing song.

Otherwise you have to press the big red "Transform" button at the bottom-right of the dialog to start the merging/transformation process with the selected options.

Previous Topic: MIDI Options
Next Topic: Machine transformation

Machine transformation options

With a complete and correct machines.ini you can manipulate the data of a single generator machine of an existing song. First you have to choose the input file in the previous tab, than press the **Get generators** button to get a list with the names of all generator machines in your file and select the machine you want to process. Remember to do this every time you choose another input file, to get correct results. Leave **none** as source selected, if you don't want to do this machine specific stuff, especially if you want to use **blow_up** or **repeat** from the previous tab.
These options, with exception of source machine transposing, only work for multi-track generators.   You can't clone the notes of for instance an Arguelles Guru...

transpose (src)
You can transpose all notes of the selected machine.

Rip ES-9 data
The ES-9 machine needs additional data, which can be edited in BUZZ. If you choose such a machine from the list and check this option, the data is saved to the file **es-9.dat**, overwriting any existing (so keep backups of your favorite settings). This file will be used later for conversion or cloning to an ES-9 machine. Take care with this, as a lot of ES-9 settings seem to freeze the BUZZ sound output, you may have to reload the song several times or even to restart BUZZ, to hear it again. In worst cases you have to fix the ES-9 data in BUZZ or to remove the machine. This machine probably is still in an experimental state, as is its editor.

destination
If you choose **none** here, you may only transpose the selected machine above. If you select one of the generators, a new machine will be created and all note and volume events of the source machine transfered to this new machine.   You get a corresponding pattern for every source pattern and the same sequence for the destination machine, you have only to connect it to the master in BUZZ.
If you select **tracker** as destination you can choose the number of the wave instrument to use. If the source also is a tracker, even the effects and tick subdivision values will be transfered. With a wave instrument set to zero, the original sample numbers will be used
(probably only useful for transposing).

New in v3.6: Look for info regarding the generator list

transpose (dest)
The notes of the newly created machine will be transposed.

Previous Topic: Transformation/Merging Options
Next Topic: Wave Transformation Options

Wave transformation options

Tune wave
In BUZZ you can change the root frequency only between 11025 and 44100, this is only a limitation of the input routine, if loaded other frequencies are recognized as well. So here you can set the frequency as you like, just select the wave from the list (and the number of the layer for multi-sampled instruments).

Rip wave
With the **Rip it** option a very basic wave-ripping is implemented, as the one in BUZZ doesn't work yet. Ripping will only be done during tansformation, which may consume a lot of time, but you have to bear that, if you want the waves ;) Choose the wave number as explained above, and for instruments containing only a single sample a standard 16bit mono wave file with the correct sample rate but no loop information will be written. Its name is m2brip00.wav, where 00 is to replace with the (hexadecimal) wave number. You can import this in BUZZ and edit the loop settings manually. With non-standard sample rates you probably won't be able to play it using a wave player. Use a wave editor to convert the sample rate or additionally check the **tune** option and select any rate for the layer 1. In fact you could also just load the whole BUZZ song in your wave editor, assuming raw 16bit mono data, and if you're unlucky, the necessity of padding an extra byte before the wave data. For multi-sampled instruments the results aren't much worse in this case than with ripping: I just use the sample rate of the first layer, but append the data of all others to it, so you have to deal with them manually. It would be more convenient to save layers to several waves or to create a XI file...maybe in a later version.

Previous Topic: Machine Transformation Options
Next Topic: DOS commandline Versions

DOS commandline programs

This section may be a bit out of date, as I often change or add new options, start the program without arguments to get an overview.

Starting with version 3 the program **m2bgui** contains the GUI, the converter and the merger, no external programs are started anymore (except for the extern midi conversion). But sometimes it may be easier to use the DOS-versions (for batch-processing and the like). The GUI allows to save all options into an ini-File, which is shared between the GUI and the DOS commandline version **m2buzz** so you don't necessarily need to use the following command line options:

Usage: m2buzz.exe [options] infile.ext outfile
with ext=**mod**,**xm** or **mid** and outfile extension **bmx** or **bmw**

```
 options: (overrides ini-file +=on -=off)
   +/-l = log patterns and other info to stdout
          use +L for verbose output (effect names)
          and +Llogfile output to logfile
   +/-p/f use primifun machine instead of tracker
   +/-t/s split by track to different machines
     -w    convert no wavetable information
     +w    try to make bmw-File (XM: XI-ripping, MIDI: GM-lib, not for MOD)
     +W    include waves in output file
     ...   use +wxi_dir or +Wxi_dir to set xi_dir
!you have to set xi_dir for +w to work, best if you to do it in ini-file!
   +/-r    set XI-root transpose
   +m      intern midi-parsing (default)/ -m: extern with EasyXMid
          default instrument=channel,use +M for instrument=program
     ...   set Midi-resolution bmidi_res (default:1.0)
          (example: +m2.0 means double length,double speed buzz-file)
   +/-nn   transpose all notes (Examples: -12 or +36
   -- correct songlength (for MidimiX 1.40), unset with ++
   -cnn    restrict input file to nn tracks (channels), for instance -c16
```

Options given on the commandline override the options given in the ini-File, there is a line telling you the valid options after reading the ini-File:

```
        Actual options are: -l +W +r0 +M3.000000   -p +t
```

new:
I added some special debug options to access some functions, which aren't needed for the conversion but may help you to gather some information. The first is the possibility to dump all patterns with:

```
        m2buzz +d +dp +L myfile.bmx >bmxinfo.txt
```

This may be the final help if parts of your song got corrupted, or BUZZ doesn't load it because of other reasons. The program needs machines.ini for this, and you need to figure out yourself, what the "default bytes" (which give the dots in the pattern view in

BUZZ, normally 0x00 or 0xff) for every machine are, as this differs from machine to machine. The notes in BUZZ are stored with 4bits for the octave and 4bits for the note: 0x00 means no note, 0xff note-off, 0x01 is C-0, 0x02 C#0, 0x0c is B-0 and 0x11 is C-1 and so on, this should be easy to translate.

The second useful thing to do is finding out which layers of a multilayered wave instrument are really used (until once maybe m2buzz will automatically remove unused ones ;), this is especially useful for a MIDI drumset, where I can reduce the size of the song very much if I delete the unused layers.

```
            m2buzz +d +dw1 +L myfile.bmx >bmxinfo.txt
```

will produce a table for the first instrument in the output part between the PATT and the SEQU section, where for every note with 1 or 0 is indicated, if it is played or not. Choose another number instead of 1 for another instrument.

b2buzz is the DOS version of the merger, start it with

```
         b2buzz -?
```

for instructions...

gus2xi is used to create a complete patchset from a set of (classic) GUS patches and a correct ultrasnd.ini. Just start it with

```
         gus2xi ultrasnd.ini
```

and all patches will be created. Take care as existing .XI files will be overwritten without warning. To use these patches you currently should transose your song an octave down and correct the drums an octave up. Also the **remove loop-flag for drums** option should be checked. More details are given in the next help-section.

Previous Topic: Wave transformation
Next Topic: How to create a GMIDI XI-patchset

How to create a General MIDI XI-patchset

The MIDI file fomat just gives you the notes and via the patch number an additional hint, how the instrument should sound. With the General MIDI standard this hint is a bit more commanding, but nonetheless the used sounds depend on your MIDI device. Today's soundcards commonly use sample data as base for their sound creation, the samples and parameters for one single sound often is called "patch". This data is stored either in the cards wavetable ROM, wavetable RAM or with most recent (PCI) cards even in system memory. If the sounds aren't stored in ROM only, they can be changed or replaced, so even with the same card a General MIDI song can sound very different. Also the amount of memory and the format used to store it differ widely.

With m2buzz you have three options to deal with this:
1.Ignoring instruments: The first is to ignore all information about the instruments and convert the MIDI file to a song with tracker machines without instruments, any dummy instruments or to any other generator than a tracker. After conversion you can change the used instruments in BUZZ.
2. Using vMidiOut: The second way is to rely on the MIDI capabilities of your soundcard, and use vMidiOut as conversion target. All patch numbers of the original MIDI file will be retained, and with exception of the things m2buzz does not support (controllers, tempo changes, sysex, timing resolution) the converted song should sound as the MIDI file played through your sound card. Of course the disadvantage of the MIDI format also remains: the BUZZ song would sound as different on other computers as the original MIDI file, so I don't think this method is adquate for distributed songs.

3.Sample patchset: So the third method, which shall be discussed here, comes into mind: Just use the patches of one soundcard and convert it in a way m2buzz and/or BUZZ can deal with them. This could be done with a special General MIDI machine for BUZZ, which would include all the samples (with or without the option to change them), but I use the tracker machine and include the samples in the song instead. This increases the songsize, but allows more freedom in choosing the instruments without loosing the reproducibility of the sound on other computers. As intermediate format I chose the Fasttracker XI instruments, as they already were part of the XM support in m2buzz. As patch source format I had to decide between the Gravis Ultrasound (GUS) patches and the Soundblaster AWE32/64 soundfonts (SBK/SF2), both cards have wavetable RAM and hence a lot of patches freely available on the net. I finally use the GUS patches, because I found them simpler to decode and not depending on any ROM samples as a lot of soundfonts for the AWE.

With **gus2xi** it should be easy now, to create a complete patchset. Get any complete patchset for the classic GUS cards, for instance from the CubicPlayer homesite:
http://www.cubic.org/player/
The patchset usually has a size of about 5-10MB. There should be a bunch of .PAT files and a file ultrasnd.ini (if the last isn't there, get it from the MAZ pages:
http://www.maz-sound.com (in the "Players" section under the topic "Open Cubic Player")
Edit the paths in the ultrasnd.ini (PatchDir=...) to point to the paths, where your PAT files

are (only the sections "Melodic Bank 0" and "Drum Bank 0" are used).

Now extract **gus2xi05.zip** (should be part of the m2buzz-package) to the XI subdirectory of m2buzz. Copy the edited ultrasnd.ini there, too. Now start from a DOS-Prompt:

```
        gus2xi ultrasnd.ini
```

and a lot of XI files should be created (if not, the paths in the the ultrasnd.ini probably are wrong). Take care as existing .XI files will be overwritten without warning.

Your XI path in the MIDI section of **m2buzz** should already point to this directoy, so you are ready to go now: Just change the instrument option from "use channels" to **"use programs"**, and make sure **"multi-sampled drumset" and "drums follow pitch" are NOT checked**, but the **remove loop-flag for drums** option should be on. Also the **"pitch/drums correction" should read -12,0**.

Before conversion of a MIDI file you should also get used to the concept of the MIDI resolution, as only with a fast BUZZ tempo you can get an adequate conversion of MIDI timing. This is no limitaion of m2buzz but of the (BUZZ or MOD/XM...) pattern format with a row as (almost) minimal time difference between events on a track. So set the value in the MIDI section to a value of at least the default **16 TPB but even better to 24 TPB**. Let the BPM-resolution on 1. With the BUZZ default of 4 TPB you'd get awful results instead.

With both variants (2 and 3) all notes and patches should be converted correctly. NOT converted by m2buzz are all controllers and tempo changes in the song, that is only the BPM rate at the start is used. GM generally is assumed, but of course you can change the instruments in the XI directory as you like.

Previous Topic: Commandline Versions
Next Topic: Missing Features / Bugs

Known Bugs / Missing Features

The used compliler supports Windows NT (as its name is RSX-NT), but there were errors in my program, which only triggered in NT. They are fixed now, but as I don't use this operating system, I can't tell for sure, if they all are gone or maybe some new introduced. So regard this as less supported than Win95 (the same applies to Win98).

As of version 2.1 you can listen to a lot of the converted songs with pleasure now (I hope at least ;). There's often some noise and crackle. And there's this list:

Known Bugs / Crash reasons

There's a strange bug if you convert a song with tracksplitting to primifun machines. After conversion the song plays allright, but if I remove two machines, BUZZ crashes.

BUZZ may also crash, if the sample map of a XI-instrument is numbered descending, look below in Missing Features/XI for details.

Because of incorrect or only unusual input files the converter sometimes goes into an (almost?) endless loop, reading EOF, in the worst case filling up your harddrive with an evergrowing logfile, this should always be checked, but isn't done yet. (for instance: invalid patterns in XM when given patternnumber higher than actual patterns in file).

Currently the new transformation options are introduced and tested only one at a time. So you should only do one of the **cloning** or **blow_up** or **repeat features.** If you want to apply more than one of these things, achieve it with several steps. Otherwise the output file may be incorrect and may cause BUZZ to crash. This may be fixed in later versions.

Missing Features

General

If the maximal allowed polyphony for a machine is overwritten, the conversion exits with an error message. An additional splitting into at least two machines would be fine, but I doubt I'll do this. This concerns especially the "no split" option for more than 18 tracks (with the tracker machine, different limit for others), and MIDI files with a high polyphonie in a single track. Try to split up the latter with a sequencer.

The resulting songlength always is truncated to 112 patterns (BUZZ song). If only a part of the patterns is used for every machine, this value could be increased, because only the used patterns are included. But I guess this is no severe limitation to anyone.

MOD / XM

Many module effects are implemented only rudimentary, often the parameters are wrong (portamientos, vibratos, note delay, set volume). I tried to have correct timing (set speed,set bpm) and pattern sequence (pattern break is implemented, but for pattern break with offset I only allowed one offset value for the following pattern), but pattern delay is still missing.

No volume column effects of XM's are supported.

If a XM-file has Amiga frequency table note conversion and other frequency related things (samplerate, pitch slides) may be wrong.

XM / XI-instruments (for MIDI)

Volume information of instruments or samples is used now, but in BUZZ it's for the whole instrument only, while XI's allow volumes for every sample, which only could be achieved by scaling the sample data. Instead simply the volume of the first sample is used for the whole instrument.

Support for volume envolopes has begun, but may have errors.

Panning envelopes currently aren't useful in BUZZ.

Ping-pong loops aren't supported by BUZZ, it could be achieved by expanding the samples, but not done (yet?).

For multisampled instruments a monoton ascending sequence (0-1-2-...) of samples is assumed, as normally should be the case. If in contrary the original instrument had a samplemap like 0-1-0-2, in BUZZ you would have 0-(1-1)-2. If the samplemap is something like 2-1-0, BUZZ may even crash if the instrument is played. I did no reordering (yet?) because only one of my files exposed this behaviour and it would need some reorganization of the program.

MIDI

The resulting BUZZ song is truncated to 112 patterns regardless of the Midi resolution, that means the higher the Midi resolution the shorter the converted part of the song.

Speed setting is only used at the beginning of the song (time=0), no set speed changes during the song are applied.

Because of the limitation of one event in a row of a track there's no perfect way to get notes shorter than that correctly. The best is to increase midi_res. The note delay command seems not work for note offs in BUZZ (even if done with volume 0), and a second command on one row (like note cut) can't be done. Additional tracks to avoid overwriting of events in the same row are allocated automatically now, although not more than 32 for one machine. You get warning messages if this is insufficient and also about the short notes.

No controller commands are converted, pitch wheel and volume would be fine.

Merging/Transforming

The blowup doesn't change tracker effects, so everything using the tick division will be wrong. There are other machines affected (gapper), which operate with fractions of rows.

Single-track generator machines like the Arguelles Guru can't be cloned yet, because I assume a conversion track by track.

If you merge two songs including the master patterns, the resulting number of master patterns should be restricted to 112.

Previous Topic: How to create a GMIDI XI-patchset
Next Topic: Tips and Comments

Tips and comments

MOD conversion: I never intended a 1:1 conversion of existing modules, and I doubt this can be done at all, at least with the current tracker machine. If this is what you want, get somebody to write another MOD player/tracker machine with some kind of BUZZ interface, but not me. I'm very satisfied with the current state, taking into account I primary only wanted MIDI input into BUZZ and I guess it's already a big help if you want to refine your old or new modules.

MIDI conversion+merging: you should know before, if you want to add a MIDI voice to a song. Choose a high BPM-rate in this case (I use about 500 and midi_res=4). You can also use the blowup feature now, but it has some limitations.
The synchronising of the new voice to the older part is very tricky, I start trying it in the sequence editor, but than use the MIDI time shift option. This has the advantage to let the patterns start with for instance 64 line boundaries, so they better fit with the song and allow shifting and the like. The long time sync seems poor, better use several shorter parts. The time scale option is intended to fix this, but I don't know if there is an universal value for this, didn't try too much.

Nice thing: I very much like conversion of MIDI files to the tracker, the organ and plucked string and finally merge everything together and pick out the best voices. This easily can eat up a lot of processing power, but with some additional effects the results may be really worth it (remember to use a midi_res of 4 or above). In fact this is the reason I published this version that early: I wanted to release some demos of this, but they are all much too large, even without samples, so I decided to release the program instead :)

Beta state: All conversion functions should work as in version 2.1, apart from the (many) bugfixes (and the maybe new bugs), if you experience strange results it often helps just to restart the program, although I hope I've caught all variables for reinit now. With merging this may be different: I've got a lot of useful results, but I need input from others about bugs or for improving fuctionality. I know the window output handling doesn't look that perfect yet, but it works...

Filenames: I know I should check files for overwriting, and ask you about. But without this and for easy loading into BUZZ I recommend you to leave the outputfilenames unchanged and save them from BUZZ with another name. This way you'll never loose some unrecoverable song and you can have the converter active with BUZZ and always load m2buzz.bmx or b2buzz.bmx from the recent files. I've found this to be most convenient.
Take care also with the extern converters it2xm and mod2xm, you'll loose any existing XM with the same base name as the IT/S3M/MOD input.

Backups and if they are missing: As with every other computer program also with BUZZ the golden rule is: backup your work often with different filenames and to different places, use floppies, give your songs to friends, burn them on CD... Remember, BUZZ is alpha, and I had a lot of difficulties with programs which were sold in major versions much greater one. With BUZZ and also with m2buzz you have the opportunity to overwrite some months work in parts of a second, so reduce the risk, where you are able to. The m2buzz transformer allows input and output to be the same, but if something goes wrong you may loose, so better use different names. I only had trouble with BUZZ saving (additionally to the usual startup crashes) because of the paramter saving thing: you find a good setup for a machine, save it, decide to make some patterns for the machine, play around with it, save it again and whoops: your original setup is gone, as always the current parameters are saved as track setup. But I've got mails from someone, who said he saved a song normally and wasn't able to load it again, another getting into the trouble because BUZZ crashed during save. In these unfortunate situations, where the song at least partially should be there, but you can't load it again into BUZZ, the commandline "m2buzz +d +dp +L myfile.bmx >bmxinfo.txt" may help you, if you think your song is worth the effort to input all the data again. If m2buzz doesn't return, because of the corrupt song, Ctrl-C may help.

Split by instrument: I guess this was the most wanted option (if not the only ;) for the converter, so finally here it is (Here were mentioned a lot of things, which didn't work with this option, but I hope, these are all gone now ;). Removing duplicate patterns makes this option even more useful now, but I'm not sure if I'll implement a feature allowing to collect several instruments into a single machine again (some kind of instrument grouping).

Previous Topic: Missing Features / Bugs
Next Topic: History

History **revisions** (m2buzz.rev could be more up to date)

1st public 98/07/12
-just a mixing of the module disassembler by David Lai (thanks) and an 1tracker 8channel input buzz song
-note and sample info, correct sample remapping
-effects: partially note delay, pitch slides
-no tempo information at all, no pattern break

internal 1a
-almost completely replaced input buzz song with bmx-format definition by Oskari Tammelin (thank you)
-rewrite of original diskoriented aproach to memory oriented aproach: all pattern information for the whole song is now stored in memory (necessary for multiple machines, but possibly difficulties with large modules (sample size doesn't matter) especially with my old 16bit DOS-compiler (other compilers: check misc.h for int and long sizes, portable types maybe not everywhere implemented)
-split mod-channels to different machines

internal 1b
-implemented XM-loader from scratch, still no buzz conversion
-add modul-loader from scratch, almost complete replacement of original module disassembler, stripping of all unused code (sample ripping,...)
-reintegrate buzz conversion for modules and made default option: now it's really mod2buzz :)
-implement sequence and pattern information for master machine

internal 1c
-restrict channel number (for easyxmid)
-integrate easyxmid (start it, scan for channels, convert xm)
-pattern break
-primifun maybe broken
 -gcc compatibilty (allows larger patterninfo, run with win95 or cwsdpmi-package)
-1c ot-release for Oskari Tammelin

internal 1d
-Win95 port (gcc, RSXNT): converter as windowed console (m2bwin) and a GUI for starting it (m2bgui), my first Windows program ;)
-strip empty song positions off sequence section
-correct note-offs (always in the same track and with the same note as notes ons)
-midi-parser (using midilib)
-trial bmw-support

internal 1e
-reading of XI-files and XM instruments (not passed over to BUZZ yet)
-changed commandline opts (to set defaults independent from gui)

-now conversion of XM with only single sample instruments with roote C-4 seems to be correct (means: s3m->ft2->xm->m2buzz +x +w->buzz)
-corrected bug with XI's with no samples: XI-file is written (probably incorrect), but instrument is not included into bmw anymore, no remapping at the moment (is correct if all those instruments are at the end, as mostly the case)
-control of patternrange -> "--" not necessary needed anymore

public v2
-GUI options save and load, ini-file handling
-reworked GUI to support all options
-documentation v2 /"help file"
-some GUI bugs removed (path handling, options)

internal 2a
-hups, forgot/dropped the mod sample mapping, reimplemented, also done for XM now
-MOD and XM -> BMX with wavedata included ;)
-changed wave options: dowave 0) no sample info 1) extern samples: no MOD; XM,MID: =dobmw   2) intern samples   (default)
-read wavetable info from XI (for MIDI): gm_00.xi...gm_09.xi for channels
-track program changes: GMIDI-lib gmp_00 (drum) ....gmp_80.xi
-finally multisampling,fine tune and transposing seems allright (really always?)
-little bit better volume handling (set volume *2)
-some crash opportunities resulting from incorrect input files removed

public 2.1
-adapt GUI to new options
-helpfile finally in Winhelp format
-changed midi_res default to 2.000
-channel instruments now named gmc_01.xi...gmc_10.xi
-limit patterns for midi-files to 112, BUZZ-songs should handle more than 112 songpositions, but I couldn't get it to work
-for BMX+waves changed sample remapping: now dummy samples are used to replace empty samples and instruments, so all instrument text messages remain visible ;)

internal 2b
-developed partially functioning buzz merger b2buzz as new project
-added binstr_add for better merging results (option -a)
-finally got the tabbed dialog for the GUI working, but the 3D input line look is gone :(
-integrated some merging options into the GUI
-XI-file conversion into empty song with 1 instrument
-some Midi handling improvements:
 -patches now only include drums if used
 -better Midi-track handling for Note-ons (use empty tracks, if any, to avoid overwriting of events)
 -note-delay for midi, ticksubdiv=15 (not used by BUZZ for note-offs?)
-added commandline options -x* -x+   -xa(=-a) for merging

-portamiento/2 (why was it doubled?)

internal 2c
-integrated merger into GUI-version
-day of the bugs: why did version 2 work at all :] fixed memory corruption and used patterns handling
-introduced machines.ini and got alot of machine parameters
-no merging options during conversion anymore, position scaling and instrument shifting done alone with machines.ini instead: more obvious for the user
-integrated converter into GUI-version
-adapted help-file and GUI to this
-first try for sample/instrument volume handling
-now we read MIDI-files 3 times: 1) get timing 2) get polyphonie+extra channels 3) convert
-use real note offs instead off volume 0 (I didn't find this until now ;)
-extra option to choose if to use note delays (because no delay support for note off :(
-output to organ and some other machines :) some additional data is put into machines.ini, but no general method to include a new generator (problems: parameters
-may be omitted--, data init)

public 3beta
-threading in GUI version
-added blowup option, now it becomes the: Buzz trans-former m2buzz ;)
-you can omit second file to transform first only
-added commandline options for b2buzz
-Bugfixes:
 -reinit global data in GUI: note_map
 -allow some non-standard MIDI files to work (what data lies behind tracks?)
 -create ini-file (-i) for commandline to allow DOS only conversion (but why, BUZZ is Win95 ;)

3beta2 (3a)
-split by instrument :) this uses an additional pass and only uses info from the current pattern, so you miss some events, where the instrument number isn't given in the current
pattern :(
-removed unused patterns (not much size gain, but better for editing)
-integrated it2xm by Andy Voss as extern converter for IT 2.xx (?not 2.14+)
-fixed XM patterns for this, where patternsize!=given patternsize(maybe this was also the bug with files saved by modplugtracker)
-added some DOS commandline options
-implemented maxpatterns=112 also for mod/xm (nice for converted midi-files)
-implemented maximal tracker polyphony (seems to be 18, set in machines.ini), this probably was the reason for a lot of former crashes
-fixed split by instrument (should now work as intended and without the above mentioned limits):

-keep last instrument of every track*pattern
-set note-off in old instrument if change to new
-changed pitch for mod conversion: notes 2 octaves up, frequencies/4, so that notenames are equal to those in FT2

3beta3 (3c)
-option -dp to dump pattern data for DOS-version
-option -dw.. to dump used notes for a wave instrument (to allow to find out unused layers)
-MIDI: -configurable number of patternlines (increase for greater possible songlength)
-option to start time (and patterns) with first note-on (better for synchronizing)
-Transformer:
-sequence multiply =double song (append all sequences a second time)
-transfer notes/volumes of one machine (patterns,sequences) to another machine (other or same type!)=machine cloning +transpose, select wave for tracker
-transpose a machine
-put all this into a new tab in the GUI
-started with volume envelopes
-tune wave option (allow to use frequencies <11k or >44k)

public 3d
-envelopes:
-env length set to 324
-(more) correct fadeout to release conversion
-rip ES-9 data and use ES-9 as generator (convert/clone)
-fixed NT problem with XI (and hence MIDI) conversion
-fixed transform/merge bug with files containing empty sequences
-new: zero-loops for click removal with non-looped samples
-new: count start zero option
-MIDI: option to use splitted drums (gmp_81.xi...gmp_ff.xi)
-fixed bug with MIDI program change (was not correctly used if more patches in same channel)
-option: start BUZZ from GUI (because BUZZ crashes that often :(
-fixed merge bug with files containing empty waves
-handling of missing XI (at leat for MIDI conversion with patch change)
-added 4 new Jeskola machines to machines.ini: Crossdelay,Reverb2,Bass3,Filter2
;) now we have Buzz 1.0 and even a new tracker (with less clicks),
so the zero-loops should already be obsolete, but m2buzz hopefully not
-included first version 0.5 of **gus2xi** for patch conversion
-changed default to several drums instead of multi-sampled drumset
-added some options for gus2xi-patches (drum transpose,follow, no-loop)
-added Wave-rip option (very basic only)

public 3e
-added midi_tpb (enhances/replaces midi_res): defaults are midi_res=1.0 and midi_tpb=16, the total resolution (=former midi_res) is now midi_res*midi_tpb/4

-stripped the empty master sequence for MIDI conversion
-added vMidiOut and Bass-3 as possible conversion targets
-changed conversion from MIDI to vMidiOut to include channels and programs:

- o.k. for MIDI-files type 1, worse results for type 0
- choose the following MIDI instrument options for this to work:
  "use programs","drums follow pitch" and "pitch correction: 0,0"
- as usual for MIDI conversion use a TPB value of 16 or above

-option to set vMidiOut MIDI device
-remove duplicate patterns
-use external mod2xm by Arnaud Hasenfratz for S3M-files
(and optionally for MOD)

version 3.5

-added contest_number option for IBMC (b2buzz commandline only)
-fixed BUZZ reader to allow any machine data size (VST_loader and WaveAss bug)
-changed compiler again (VC seems only compiler for BUZZ machines
and player code?) ->fixes for _beginthread and return a+b (speed optimizing)
-added play option for input modules (uses BASS lib by Ian Luck)
-added play option for BUZZ songs (uses buzzplay code by Jeskola),
(seems not to play all songs and casually crashes the programm or PC!)
-changed blow_up to change TPB instead of BPM
-changed optical outfit (especially to put some more attention to the transformer/merger)
-full/sequential merge option: this takes the complete sequences of both songs (including the master connections/sequences of the 2nd one)
-two additional setup patterns for the master to mute song A or B and set initial tempo, master volume,...
-create a second machines.ini (-> machine2.ini) from DLL if playing a song or missing a machine during transforming/merging
-get BUZZ directory from registry
-bugfix: the merger always was wrong if there were connections in the second song :(
-prefix "A " and "B " for machines and master patterns after merging
-wave shifting for other generators than tracker (drummer), in stats (pro2) and effects (vGraphity)
-source transposing for single track generators (guru,pro2)
-warning for master patterns >112

version 3.6

- fixed MOD sampleoffset effect (thanks to Renegade)
- fixed MOD volume handling and volume slides (thanks to Daniel Baum)
- allow (almost) any generator as conversion/cloning target; exceptions are generators without note usage (drum + special purpose machines...) , even trackless machines (Guru,Guru2, Pro2) are now handled as single tracked
- wave-inclusion options now independent from conversion target machine, so alternative trackers (like the Arguelles Pro series) get a chance
- fixed and improved collecting of machine info (it also tries to find the volume column), machines.ini-file parser (init/empty machine data length now up to 50 bytes, so

M4 and co work) and machines.ini contents (added Arguelles Pro4 for future purposes, removed some superfluous machines which are now handled correctly from machine2.ini)

version 3.61

- bugfix for converter crashes

version 3.62

- increased machine data maximum again (->Guru 5)
- removed 112 pattern/sequence limit (Jeskola removed this from BUZZ with some latest 1.2beta, take care with older BUZZ versions)
- updated BASS-lib to BASS 0.8 (-> Win2k compatibility)
- changed conversion of MOD effect 0x03 (TonePorta) to Jeskola Tracker effect 0x05 from *2 to *lastspeed (=MOD speed, default 6)
- new conversion of MOD 0x05,0x06 to Jeskola Tracker 0x05,0x06 with last effect, VolumeSlide ignored
- Matilde-Tracker effects implementation

future?

- correct the problems with gus2xi
    - instrument grouping
    - MIDI:speed changes,controller conversion
    - conversion to MIDI files (especially vMidiOut)
    - GUI: some kind of scrolling for the logwindows and 3D look again
    - Merger/Transformer:
    - replace sample x with y
- "MIDI-controller into column"
- some module effects to be done/corrected:
    - portamientos (amiga freq-tables!)
- tempo change correction (especially easyxmid), note delay
- note delay correction
- pattern delay
- correct volume handling

- ping pong loop expanding

Previous Topic: Tips and Comments
Next Topic: Contacting the author

Contacting the author (that's me I guess)

I often mentioned myself in this document, so if you want to send comments, corrections, bug reports or suggestions, do this to
muehle@theory.phy.tu-dresden.de
and as you see, I'm german, so don't laugh too much about my english, or at least, don't tell me.
Latest revisions and some bug/progress reports currently are to be found at
http://rcswww.urz.tu-dresden.de/~vmuehle/m2buzz.htm

Previous Topic: History
Next Topic: Legal Stuff /Thanks

Legal Stuff

The author doesn't take any responsibilities for anything (good or bad) resulting from use of this code/program to any person, data, computer or foreign galaxy. While you can earn your money with converted modules, please don't take me mine accusing me, I had done something mean to you. I tested my program, to reduce damages resulting from it, but I only did this alone on one single computer with finite input constellations, so you should assume that

THIS CODE IS NOT CAREFULLY TESTED!

at least in comparision to commercial programs or programs with a larger user base.

The resulting files sometimes crash BUZZ, although I tried to reduce this. I can't made responsible for damage or data loss, neither in this case.

The bass player library is done by Ian Luck, the BUZZ player by Jeskola. Both are free for freeware applications.

All trademarks named in this document are trademarks of their respective owners.

Thanks

Big thanks to Oskari Tammelin (Jeskola) for BUZZ. Never has tracking been that lot of fun. Let's grow BUZZ.

Thank David Lai for his source code (moddis) and Stefano Maini for Easyxmid, giving me a very fast way of doing what I wanted (although now almost everything is replaced by my own). The intern midi code is almost unchanged based on the midilib by Tim Thompson, very fine package.

I couldn't get the propertysheet page to work, so I used a tabbed dialog (c) Edward McCreary (1994). The original code was for Win3.1, I did the needed adaptions to Windows 95.

I don't intend to take the troubles of IT conversion on me, so I'm thankful to Andy Voss that somebody already has done this. The same to Arnaud Hasenfratz regarding S3M conversion with Mod2XM. Nice that there are some people who write converters ;)

Further I wish to thank

- -Rasmus Schultz for his very constructive help
- -Gabriel Winter for helping to fix the NT and other bugs
- -Nick Alevridis and Fabio Napodano for help and suggestions

and last but not least MAZ for hints, discussions and his general work.

Previous Topic: Contacting the author

Return to Contents

MOD-files and trackers

The Commodore AMIGA had an in it's days revolutionary 4 channel 8 bit hardware mixing sounddevice, while the PC Soundblaster era was still far away. By simply adressing the four channels of the hardware it was possibly to play tunes with almost no processor power used (yes, I had multitasking with music playing in the background and a texteditor and a compiler running on a 8 Mhz machine with 1MB RAM :) so the idea of 4 channel MOD-files or simply modules was born. They generally contain first some header information, than all notes to play packed together with other effects to apply in patterns of originally 64 rows, and finally the samples to use as instruments.

The programs to create such fine tunes were called trackers, with, i guess, Protracker one of the first and widest used. Already on the AMIGA it was tried to break the 4 channel barrier with software mixing, take Octamed as an example, but on the PC there was never such a limitation: either you had only one lonely channel (or two, if you look at stereo this way) or later there were about thirty independent channels with wavetable devices as the GUS or AWE are (only recently you also get cards with independent wavedevices on the PC).

So it came, that short time after finally the first MOD-players for the PC came along, trackers with more channels were used and often let sound the old 4 channel tunes a little bit pale in comparison (not always of course, musical genius doesn't depend on the tracker used ;). I remember Fastracker (I) with 8 channels, Multitracker and finally the famous Screamtracker with 32 channels and a bit later Fasttracker II (somebody called these trackers of the second generation).

Most recent developments I'm interested in are Impulse Tracker (with 64 channels + virtual channels), Modplug Tracker (bringing all these trackers to Windows) and of course BUZZ.

XM-Files and Fastracker II

While Screamtracker was first and is loved because of it's keyboard-only navigation, Fasttracker II gained it's acceptance because of it's mouse driven GUI   and one great thing: MIDI in and output.
Advantages in the song format "Extended Module" (XM) are use of 16Bit samples and multisampled instruments with volume and panning envelopes, which may also be separately saved as XI files. This instrument format is additionally used by the converter to supply the patches for the MIDI conversion and you can convert a single XI file into a BUZZ song (without patterns) and merge this into an existing song.

Screamtracker S3M files

For conversion with m2BUZZ, you can load Screamtracker S3M files with Fasttracker II and save them as XM. These XM files (as XM files produced from MOD files this way) have a so-called AMIGA-frequency table, which probably is supported worse by m2BUZZ than the Fasttracker II native linear frequency table.

I abandoned the plan for direct S3M support by m2buzz and just do it by another external converter now: mod2xm by Arnaud Hasenfratz, which seems to do the job, although I didn't test it too much.

XI directory

This directory has several tasks, which are important for the correct conversion of wavetable information (Instrument/Wave options), so it's path should be properly set:

-During conversion of XM-files with checked **Make BMW** option all instruments of the XM-file are ripped to this directory.

-For internal conversion of MIDI-files with either the **BMX with Waves** or the **Make BMW** option checked, the instruments to use for note messages are looked for in this directory. Depending on the MIDI-options either a set of 16 files gmc_01.xi...gmc_10.xi (**use channels**) or 128 files gmp_01.xi...gmp_80.xi (**use programs**) plus drums (gmp_81...gmp_ff or gmp_00.xi) should be here.

-For XM-files and MIDI-files with checked **Make BMW** option the path to the single XI-files in this directory is written to the BMW-file, these instrument files should still be here, if you want to load the song into BUZZ.

-You may include this directory in the BUZZ-wavetable paths, but you should beware of the BUZZ a15 wavetree bug with too many instruments in one directory.

BUZZ

BUZZ is a great composition tool with a big heritage from trackers, but much more features (Introduction) It is written by Oskari Tammelin, and in it's present state (as of august 1998, version alpha 15) already very usable. The main feature still missing (at least for me)   is complete MIDI-support, and there are of course still some bugs and glitches. According to messages in buzz-talk@unreal.org the author is currently working on BUZZ 2, but there isn't known too much about it yet.
Keep up the good work and good luck in all your doings, Oskari.

BUZZ Songs: the BMX and BMW formats

All information of a song is stored in one of the two BUZZ song formats. They consist of several sections and start with a header containing pointers to every section. Currently these sections are: MACH where all used machines are introduced, CONN giving the connections between them, WAVT containing wavetable information, PATT with the patterns for every machine, SEQU with the information of the sequence editor and BLAH as the text section. Finally the **BMX** format only has a WAVE section, where all wavedata is stored.

While the **BMX** format is completely containing everything of the song, the **BMW** format has the advantage of a much lesser size, especially if you often use the same instruments in your song or use m2BUZZ for MIDI-files. But you also get some disadvantages with **BMW**:
-You can't move or rename your instruments, because the complete pathname to them is included in the wavetable section. Although if BUZZ doesn't find an instrument and finds one with the same name in it's wavetable directories, it asks you to use this instead.
-With multisampled XI-instruments I get wrong loop settings (flag, start and end), I guess the BUZZ XI-loader overwrites some of the wavetable information in the song.
-Of course you can't delete the instruments or distribute the song this way (or you can all other BUZZ users convince to share a sample-library with you ;).

Another remark concerning the songlength: In the SEQU section are stored the songpositions and the events to occure. These can be some flags (mute,break...) or the patterns to play, stored as patternnumber+0x10. Since the highest bit isn't used, only **112** patterns 0...0x6f can be adressed with one byte. The song format allows to use more bytes for this purpose, but I couldn't achieve to load these songs with BUZZ, and a song containing 113 patterns in one sequence saved with BUZZ and reloaded also used one single byte only and hence gave wrong results. So I guess this isn't implemented yet in BUZZ, and I restricted the pattern number to 112 for both MIDI and module conversion. I've never seen an original MOD or XM use 112 different patterns, but this is of concern for EasyXMid or other MID->MOD converters as MidiMix or Modplug Tracker.

The purpose and the format of machines.ini

Most of the sections in a BUZZ song can be accessed completely with information contained in the song file. But BUZZ and hence its file format has to be open for all the new machines coming, and unfortunately the author forgot to include the lengths of some of those variable data sections into the file. BUZZ itself knows these missing parameters, because it can ask the machines (or their code) about them, but I and my simple converter don't know how to do that, and so can't access that data (which probably isn't necessary if it's a unknown machine) and, worse, the data behind.
The good news: Since the song is partitioned into several sections, other sections are not affected, and because of the known section lengths I can simply copy some of them into the new file, if I don't have to access data in them. So the machine independent merging works at least.

To improve the merging, the converter needs the missing lengths, and as they are machine dependent, I give them in **machines.ini**. Missing are the number of bytes for machine initialization (**stat_sz**) and of the initialization data for every track (**track_sz**) in the MACHines section, and in the pattern section I need the same byte numbers for the variable data for every machine and every track. For the Jeskola Tracker machine, for instance, this are the 1 byte for the tick subdivision and 5 bytes for pitch, instrument, volume, effect and effect parameter for every track. **Don't change the machine order for the master, the tracker and the next 6 generators, because the instrument shift for the tracker and the conversion rely on the correct data here!**

The version string may allow future versions to deal with old machines.ini files (but probably won't :], lines starting with ; are comments. The machine DLL name has to be enclosed in "..." because of the possible spaces, and has to start on the first column. The type is 0 for master, 1 for generators and 2 for effects.

```
version=2.5
;name of dll                       type        stat_sz   track_sz   note    instrument   volume
"master"                                0                 5          0

;Generators (tracker has to be first!!!)
"Jeskola Tracker"                 1                 1          5             1             2
3
"Geonik's PrimiFun"             1                 0          9             1             2
0
...
;Effects
"Geonik's 2p Filter"            2                 5          0
...
```

To get the numbers for a new machine, follow this procedure (or just get a good first guess by half the number of dots in the pattern editor ;): create a new song with just the master and the new machine, without connecting them, make a single pattern with only 1 track and 1 single row for the new machine, save the song and write down its filesize (**size0**). Now try to add another track ("C reate new track"). For most effect machines this isn't possible, set **track_sz=0** in this case, else save the song again (**size1**). The formula for the number of bytes per track is now

track_sz=(size1- size0) / 2

(division by two because you add the track not only in the pattern section, but also in the machine section to initialize the second track). Now delete the second track again ("Delete last track"), but add another row ("Change pattern properties", put a 2 in the rows field) and save again (**size2**). The track independent data has a size of

stat_sz=size2 - size0 - track_sz

Now add a line to **machines.ini** with the DLL name of the machine, a type 1 for generators or 2 for effects, the just calculated **stat_sz** and **track_sz** and 3 zeroes for the rest. Easy, isn't it?

This is all you need to know for merging. All the other stuff you find in machines.ini is needed for conversion only: I tried to establish a general method to allow any generator as conversion target. Although this isn't finished, you'll need a correct machines.ini now also for conversion. For the tracker only the short name as root for machine names is taken from here, other things are hard coded into the converter. But for other machines you can try to change more things, but make a backup of the file first... The number of attributes and the line with the names and values itself needs to be there, but isn't used yet. You can try to raise the number of channels for a machine, but this may crash BUZZ. Interesting may be the line with the initialization data, it starts with stat_sz numbers for the static parameters, followed by track_sz numbers to initialize all tracks. Here you may change the default sound of the machines. The other numbers represent lines without notes, leave them unchanged.

MIDI

-communication protocol between music instruments (Music Instruments Device Interface)
-allows to control music instruments like synthesizers with other devices like keyboards in form of **MIDI events**
-notes are transmitted as note on and note off events, containing note number and volume (velocity)
-exchange of a lot of other information, more or less device independent, is possible
-there are several (16) logical channels to allow better communication about instruments to use and/or to daisychain more devices and selectively adress them
-widely used standard is **General MIDI** with drum channel 10 and a patch list of 128 melodic instruments, but the actual sound of a patch depends on the used synthesizer, so for conversion a patchset has to be supplied
-together with timing and other information like lyrics everything can be stored in

MIDI-files

-there exist three types:

Type 0

all events are stored sequentially into one single track

Type 1

there exist more tracks, often one for every instrument, MIDI channel or specific controller

Type 2

more sequences/musical pieces in one file (rarely used, I've never seen one, so it's not supported by m2buzz)

see also MIDI-options

MIDI resolution

With MIDI you have a time signature for every event and can have very small delta times between events (for instance notes or note lengths); BUZZ or any tracker only allow one event per row. The time for one row is crucial for the conversion, the less the better the resolution, but you also get higher speed and larger BUZZ-files. The length of the resulting BUZZ song is restricted to 112 patterns momentary.

With a resolution of 1.000 as starting choice the BUZZ bpm-rate is equal the MIDI bpm-rate (but only BPM settings at the start of the MIDI file are processed). I often use an increased value of 4.0 - 6.0, to improve the conversion.
This means: BPM-res 1.0 and TPB 16, 20 or 24 with the new handling of this option.

There is also a connection to the polyphony of the converted machines: for every midi-track the corresponding BUZZ machine has a track number equal the highest polyphony of the track plus additional tracks to avoid that following events overwrite existing ones in the same row. These additional tracks generally will be reduced with a higher MIDI resolution.

This option is only used by the intern conversion (MIDI-options).

EasyXMid as extern midi converter

You can let my converter launch the MIDI->XM converter **EasyXMid** (by Stefano Maini) before XM->BUZZ conversion takes place, the used commandline is:

```
easyxmid input.mid mid2buzz.xm   4 10
```

(that is quantization 4, drumchannel 10). This way an uncompressed 32 track XM-file mid2buzz.xm is produced   and the resulting XM is processed with the XM converter. There is an automatic detection if less than 32 tracks are used, additional empty tracks are removed as if the **Limit Tracks** option was specified.
Generally the splitting of midi channels/instruments to BUZZ-machines is very poor and no instruments are included, so you won't hear anything after the conversion, you have to load samples manually (MIDI-options).

If you don't already have it from an older version of m2buzz, you can get it from MAZ, but the support for this extern conversion will probably be dropped soon, as the internal conversion with samples is much more convenient. Already now there's no check of the 112 pattern limit done (BUZZ song), so take care.

MAZ sound tools:
here you find everything related to tracking and more

http://www.maz-sound.com +mirrors

Generator-Lists:
Here you can chose one machine for conversion or cloning from all the generators in your machines.ini and machine2.ini (and hence in your generators directory). Machines with notes are marked with 'N', the few tracker-like machines with 'W', machines with a volume column have a 'V'.

Additional features are supported for some special machines:

-Jeskola an Matilde Tracker: conversion of MOD/XM effects
-vMidiOut : MIDI program changes are transfered to the machine
-ES-9: synthesizer setup is taken from a data file es-9.dat,
which can be ripped from any song containing this machine.

Since machine2.ini is automatically generated from all generators in your gear directory, the machines from machines.ini appear doubled in the list. This way you can transfer lines from machine2.ini to machines.ini and fix some of the things reported wrong by the machine or adapt some things to your needs (for instance the note length of the Bass machines).

MOD/XM effect conversion:

original	m2buzz	Jeskola-Tracker	Matilde-Tracker
0	**Arpeg**	-	0,=
1	**PortaUp**	4,=	1,=
2	**PortaDown**	3,=	2,=
3	**TonePorta**	5,=*lastspeed	3,=
4	**Vibrato**	6,=	4,=
5	**Pitch+Vol**	5,last3*lastspeed	3,last3 + a,=
6	**Vib+Vol**	6,last4	4,last4 + a,=
7	**XM-Tremolo**	-	7,=
8	**XM-Pan**	-	8,=
9	**SampleOffset**	a,=*256*256/length	9,=*256*256/length
a	**VolSlide**	1/2,=*2	a,=???*2
b	**PosJump**	-	-
c	**SetVolume**	Volume,=*2	Volume,=*2
d	**Patternbreak**	handled by creating extra patterns in sequence	
e0	**SetFilter**	-	-
e1	**FineUp**	-	e1,=
e2	**FineDown**	-	e2,=
e3	**GlissCtrl**	-	-
e4	**VibratoForm**	-	e4,=
e5	**SetFinetune**	b,=	e5,=
e6	**JumpTo**	-	-
e7	**XM-TremoloForm**	-	e7,=
e8	**- (ReverseSample?)**	-	e8,1
e9	**Retrigger**	e,=	e9,=
ea	**FineVolumeUp**	-	ea,=
eb	**FineVolumeDown**	-	eb,=
ec	**NoteCut**	c,=	ec,=
ed	**NoteDelay**	d,=	ed,=
ee	**PatternDelay**	-	-
ef	**InvertLoop**	-	-
f	**SetSpeed**	Subdiv+Master BPM	f,= +Master BPM

aa,bb...effect aa, parameter bb
=...original effect parameter bb; =*2 ...double original parameter bb
last3,last4...last used values of effect 3 or 4
+...do two things at once, for instance set speed and set master BPM or set two effects in Matilde Tracker