

KBD-INFINITY

Power tools for digital keyboards

MIDI Doctor Instruction Manual (Say C0h 52h)

KBD-Infinity

Albuquerque, New Mexico

E mail: info@kbd-infinity.com

Internet: <http://www.kbd-infinity.com>

Copyright 2016

Contents

1	What does MIDI doctor do?	3
2	Loading a file	5
3	Channels and tracks	5
4	The channel list	7
5	Playing a file	10
6	Editing channel voices	11
7	Writing the output file	14
8	The MIDI voice file specification	16

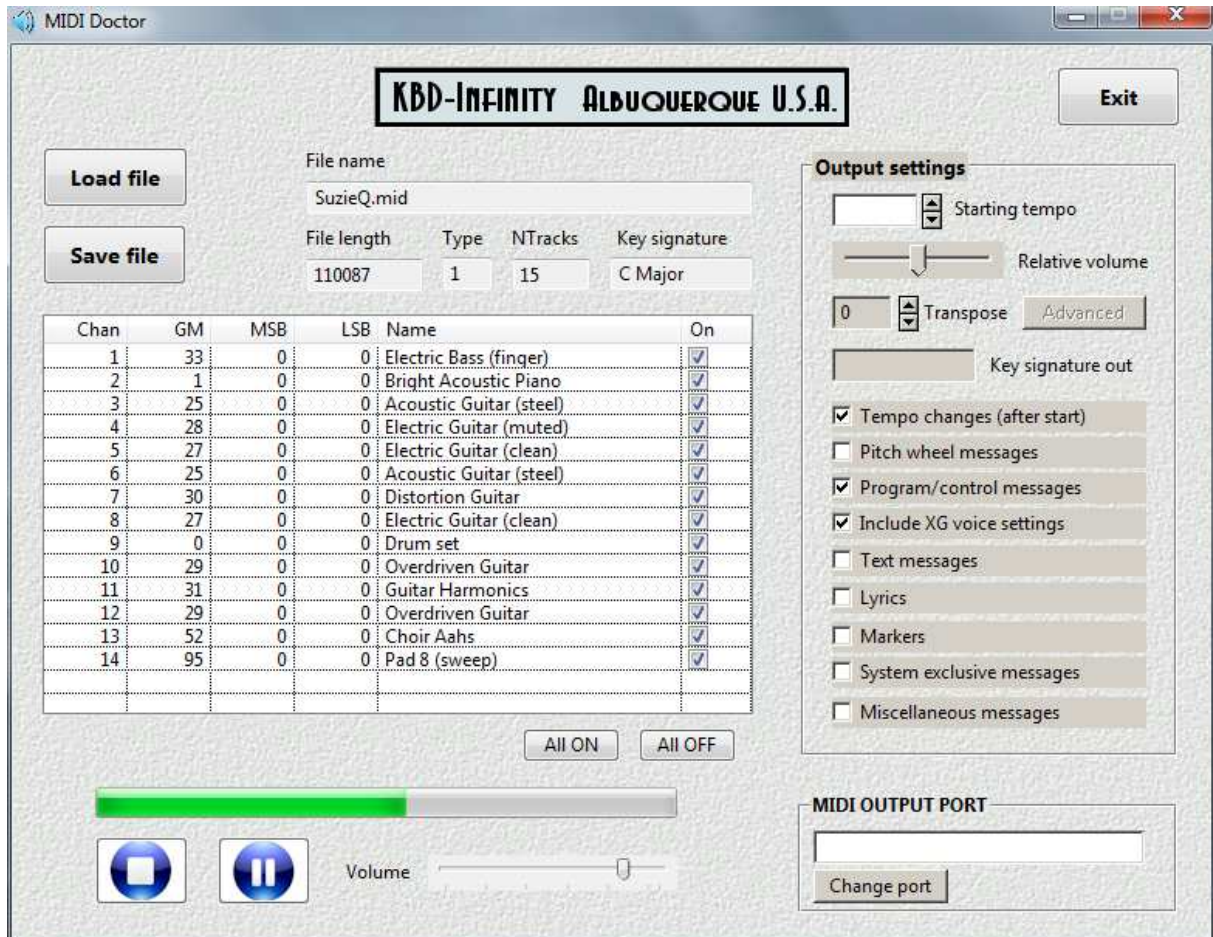


Figure 1: MIDI Doctor main screen.

1 What does MIDI doctor do?

MIDI files store music in digital format. There are thousands of MIDI files available for download on the Internet covering almost every popular song and classical work in existence. These files may have passed from hand to hand multiple times. Understandably, many files have accumulated errors or settings that are unsuitable for your application. **MIDI Doctor** creates corrected files through three functions:

- The program corrects the order of MIDI messages and stores files in a standard *Type 1* format.
- It removes unwanted or unnecessary information following your choices.
- **MIDI Doctor** provides quick path to make custom settings like channel voices (instrumentation), key signature or tempo to match your keyboard and performance.

Here are some **MIDI Doctor** operations:

- Change instrument voices of individual MIDI channels.
- Remove channels or divide channels into separate files.
- Generalize a file written for a specific synthesizer or keyboard so that it may be played on any device.
- Add XG voice specifications for individual instruments to optimize a file for your keyboard.
- Change all tempo messages in a file by relative factor to speed up or slow down a piece.
- Equalize the volume levels of individual instruments or change the general volume level in a file.
- Transpose notes in accompaniments to match the key or range of your instrument.
- Remove MIDI data that may be unnecessary for your application or synthesizer, such as text, lyric, marker, file identification or system exclusive messages.
- Invoke the full range of MIDI control messages recognized by your synthesizer (brightness, reverb, decay time, ...).
- Quickly set channel voices from INST specifications for your keyboard.

MIDI Doctor runs on any Windows computer. Unlicensed software does not save modified files. To unlock all features of **MIDI Doctor**, you can purchase a license at:

http://kbd-infinity.com/purchase_software.html

2 Loading a file

Figure 1 shows the **MIDI Doctor** main window. Press F1 at any time to view this manual. Use the Load file button (F4) to import a MIDI file. MIDI files are in binary format and come in three types with three suffixes:

- MID: standard MIDI files with instrumental musical parts.
- KAR: karaoke files with instrumental notes plus synchronized lyrics.
- STY: style files containing musical loops to create accompaniments¹

MIDI Doctor displays information about the file after it is loaded. The file name and total byte length are displayed at top left (Fig. 1) along with the MIDI file *Type* and the number of tracks. The instrument settings of the MIDI channels are displayed in the list box below². We'll discuss the meaning of *tracks* and *channels* in the next section. The program shows the key signature if the file contains a key signature message. Finally, **MIDI Doctor** displays the starting tempo (the content of the first tempo message in the file) under *Output settings*. The tempo unit is quarter notes per minute. If there is no tempo message, the program picks a default value of 120 qpm.

3 Channels and tracks

The polyphonic synthesizers of modern digital keyboards can create several instrumental voices simultaneously, each with its own set of notes. An instrumental voice is called a *channel*. A MIDI interface can control up to 16 channels, designated as Channel 00h (0) to 0Fh (16). The messages stored in MIDI files contain information about the destination channel as well as the operation that the synthesizer should perform. As an example, the MIDI NoteOn message consists of three bytes:

Byte 1: 90h + ChanNo
Byte 2: Note value
Byte 3: Volume

The quantity 90h designates the action and the quantity ChanNo is the intended channel. For example, the number 96h indicates that the synthesizer should turn on a note using the instrumental voice of Channel 06h. The second byte is an integer that gives the pitch of the note. The values, ranging from 00h (0) to 7Fh (127), encompass the 88 keys of the piano with

¹Style files contain special formatting in addition to MIDI information. Editing them with **MIDI Doctor** may destroy their utility. Use our **Style Voice Optimizer** program to set instrumental voices in style files.

²If the file contains a channel with no program (instrument) message, the channel is assigned a GM number of 00f (piano) by default.

some extras. The final byte gives the relative volume of the note. Every `NoteOn` message must be followed eventually by a corresponding `NoteOff` message (with initial byte `80h + ChanNo`).

A MIDI file contains a sequence of bytes that specify a delay followed by the bytes of a message. The delay bytes tell how long to wait before sending the message. The simplest MIDI file of *Type 0* has the form:

```
(Delay 1)(Message 1)/(Delay 2)(Message 2)/(Delay 3)(Message 3)/...
```

A MIDI player runs through the sequence, sending each message when required. With delays of 0, programs send messages almost simultaneously. For example, several `NoteOn` messages sent to the same channel at the same time initiate a chord.

In a *Type 1* MIDI file, messages are organized in *Tracks*. The term track is confusing for two reasons:

- There is no standard that determines how the division should be carried out.
- In MIDI editors, the term track is often used to denote a channel. Most editors assign messages to tracks according to their channel numbers. Accordingly, the tracks displayed in editors may bear little resemblance to the track structure of the file.

In a typical MIDI file, messages with the same channel number are often grouped in a track. Sometimes, global messages like tempo changes that affect all channels have their own track. Lyrics could be collected in a one track, markers in another and even a separate track for a single copyright message. There are two things to remember: 1) there may be many more tracks than channels and 2) the track number may not correspond to the channel number.

A *Type 1* MIDI file can always be converted to a unique *Type 0* file with no loss of information. Conversely, a *Type 0* file can be reconverted to *Type 1*, but not necessarily the same one because the original track organization was arbitrary. The output files of **MIDI Doctor** are of *Type 1* with the following standards:

- Musical messages with channel identifiers are divided into individual tracks according to their channel number. Messages with the lowest channel number are assigned to Track 0 and so on up to the highest channel number. In other words, the musical information of a MIDI file with instruments in Channels 1,5 and 7 would be saved in Track 0(1), Track 1(5) and Track 2(7).
- A single additional track contains all global and non-musical information (markers, lyrics, system exclusive messages,...).

In a save operation, all content (except that excluded user settings) is preserved.

4 The channel list

Information on the instrument voices of the file is listed in the box at left-center (Fig. 1). There is a row for each channel that **MIDI Doctor** detects when loading the file. The information is independent of any track structure the file may have. The first column lists the channel number. The second column lists the **GM** (General MIDI) number. The number refers to a standard set of 128 instruments; voices that are supported by compliant synthesizers. Table 1 shows the complete list. For example, if a channel definition includes the information **GM = 13**, then most keyboards will produce a sound like a xylophone. It's important to realize that the sound may differ considerably between different synthesizers.

The next two columns list the numbers **MSB** (most significant byte) and **LSB** (least significant byte). When combined, the numbers specify the **XG** (extended general MIDI) bank number. The synthesizers of most digital keyboards can create a set of instrument sounds that exceeds 128 in number. The extra voices can be accessed by sending an **XG** bank number as well as a **GM** number. It is important to note that the **XG** numbers are not standardized – generally each keyboard responds in a different way. This fact leads to the following implications:

- Most MIDI files that you download from the Internet have **GM** number messages only. In this case, most synthesizers make a reasonable approximation of the desired sound.
- MIDI and style files intended for a specific keyboard usually include **XG** messages. These files may sound strange on your device. The remedy is to remove all **XG** messages or change the values to match your synthesizer. The documentation for most keyboards contains a table of **MSB** and **LSB** values for extended voices.
- If you intend to distribute a MIDI file, it's best not to add **XG** messages.

Table 1: Standard instrument voices arranged by GM number.

000 Acoustic Grand Piano	043 Contrabass	086 Lead 7 (fifths)
001 Bright Acoustic Piano	044 Tremolo Strings	087 Lead 8 (bass + lead)
002 Electric Grand Piano	045 Pizzicato Strings	088 Pad 1 (new age)
003 Honky-tonk Piano	046 Orchestral Harp	089 Pad 2 (warm)
004 Electric Piano 1	047 Timpani	090 Pad 3 (polysynth)
005 Electric Piano 2	048 String Ensemble 1	091 Pad 4 (choir)
006 Harpsichord	049 String Ensemble 2	092 Pad 5 (bowed)
007 Clavinet	050 Synth Strings 1	093 Pad 6 (metallic)
008 Celesta	051 Synth Strings 2	094 Pad 7 (halo)
009 Glockenspiel	052 Choir Aahs	095 Pad 8 (sweep)
010 Music Box	053 Voice Oohs	096 FX 1 (rain)
011 Vibraphone	054 Synth Choir	097 FX 2 (soundtrack)
012 Marimba	055 Orchestra Hit	098 FX 3 (crystal)
013 Xylophone	056 Trumpet	099 FX 4 (atmosphere)
014 Tubular Bells	057 Trombone	100 FX 5 (brightness)
015 Dulcimer	058 Tuba	101 FX 6 (goblins)
016 Drawbar Organ	059 Muted Trumpet	102 FX 7 (echoes)
017 Percussive Organ	060 French Horn	103 FX 8 (sci-fi)
018 Rock Organ	061 Brass Section	104 Sitar
019 Church Organ	062 Synth Brass 1	105 Banjo
020 Reed Organ	063 Synth Brass 2	106 Shamisen
021 Accordion	064 Soprano Sax	107 Koto
022 Harmonica	065 Alto Sax	108 Kalimba
023 Bandoneon	066 Tenor Sax	109 Bagpipe
024 Acoustic Guitar (nylon)	067 Baritone Sax	110 Fiddle
025 Acoustic Guitar (steel)	068 Oboe	111 Shanai
026 Electric Guitar (jazz)	069 English Horn	112 Tinkle Bell
027 Electric Guitar (clean)	070 Bassoon	113 Agogo
028 Electric Guitar (muted)	071 Clarinet	114 Steel Drums
029 Overdriven Guitar	072 Piccolo	115 Woodblock
030 Distortion Guitar	073 Flute	116 Taiko Drum
031 Guitar Harmonics	074 Recorder	117 Melodic Tom
032 Acoustic Bass	075 Pan Flute	118 Synth Drum
033 Electric Bass (finger)	076 Blown Bottle	119 Reverse Cymbal
034 Electric Bass (pick)	077 Shakuhachi	120 Guitar Fret Noise
035 Fretless Bass	078 Whistle	121 Breath Noise
036 Slap Bass 1	079 Ocarina	122 Seashore
037 Slap Bass 2	080 Lead 1 (square)	123 Bird Tweet
038 Synth Bass 1	081 Lead 2 (sawtooth)	124 Telephone Ring
039 Synth Bass 2	082 Lead 3 (calliope)	125 Helicopter
040 Violin	083 Lead 4 (chiff)	126 Applause
041 Viola	084 Lead 5 (charang)	127 Gunshot
042 Cello	085 Lead 6 (voice)	

Table 2: Standard percussion instrument sounds arranged by note number.

InstNo	Instrument	InstNo	Instrument
035	Bass Drum 2	059	Ride Cymbal 2
036	Bass Drum 1	060	High Bongo
037	Side Stick/Rimshot	061	Low Bongo
038	Snare Drum 1	062	Mute High Conga
039	Hand Clap	063	Open High Conga
040	Snare Drum 2	064	Low Conga
041	Low Tom 2	065	High Timbale
042	Closed Hi-hat	066	Low Timbale
043	Low Tom 1	067	High Agogo
044	Pedal Hi-hat	068	Low Agogo
045	Mid Tom 2	069	Cabasa
046	Open Hi-hat	070	Maracas
047	Mid Tom 1	071	Short Whistle
048	High Tom 2	072	Long Whistle
049	Crash Cymbal 1	073	Short Guiro
050	High Tom 1	074	Long Guiro
051	Ride Cymbal 1	075	Claves
052	Chinese Cymbal	076	High Wood Block
053	Ride Bell	077	Low Wood Block
054	Tambourine	078	Mute Cuica
055	Splash Cymbal	079	Open Cuica
056	Cowbell	080	Mute Triangle
057	Crash Cymbal 2	081	Open Triangle
058	Vibra Slap		

Returning to the listbox, standard GM instrument names are listed in the fifth column. The check boxes of the sixth column have two functions:

- Only checked channels are included when you play the file.
- In saving a file, only the checked channels are included. This feature is useful, for example, to make an accompaniment file by removing the solo part.

The two buttons below the listbox turn all channels on or off.

Channels for percussion instruments have special features. Channel 09h is used for percussion by default and other channels may be used by setting MSB = 7Eh or 7Fh. Percussion instruments are generally atonal, so the second byte of a NoteOn message is used to specify the instrument type rather than the pitch. Table 2 lists the standard drum set supported by GM compliant synthesizers. Again, the standard set should be used in MIDI files for distribution. If you play a file and hear strange whistles and bells, it is usually because it is using a non-standard drum set. Changing the drum set may help. See your keyboard documentation for the GM, MSB and LSB settings for different sets.

5 Playing a file

The controls to play the currently-loaded file are located at the bottom-left corner of the window. The colored button on the left starts and stops playback. You can also use the keyboard shortcut **F2**. The **Esc** key terminates playback. Use the button on the right to pause or resume playback (**F3**). The progress bar shows the position in the file. To start a song in the middle, left-click inside the progress bar at the approximate position. As mentioned in the previous section, only checked channels in the list are included in the playback.

The main precursor to playing a file is to choose a destination for the MIDI signals. MIDI messages are simply numbers – a driver is needed to convert the number to an appropriate audio signal that is sent to computer speakers or an external device. You can choose a driver with the **Change port** button at lower-right. Here are some possibilities:

- In the first time run, the device is the **Microsoft MIDI Mapper**, a **Windows** default. This is a rudimentary driver that approximates the set of **GM** voices and sends the result to the computer speaker.
- For a better sound with lower latency, download **CoolSoft VirtualMIDISynth**. This blog article describes the procedure:

<http://kbd-infinity.com/kbd-infinity-blog/replacing-microsoft-gs-wavetable-synth/>

- Install the **LoopBE** driver to send the output of **MIDI Doctor** to the MIDI input port of another software program on the computer. One possibility is **VST Host** where you can experiment with virtual instruments.
- You'll probably get the best sound by using the synthesizer of a good quality keyboard connected to the computer.

6 Editing channel voices

One of the primary functions of **MIDI Doctor** is tuning the voices of individual channels to change the arrangement of a file or to optimize it for your keyboard. To adjust a voice, double-click on a row of the channel list. **MIDI Doctor** opens the dialog shown in Fig. 2. The top controls show the current settings of the **GM** number and the **MSB** and **LSB** values of the **XG** bank number. You can change these quantities. Use the set of controls on the left-hand side to customize the voice. It is important to recognize that some of the controls may not have an effect, depending on the sophistication of your synthesizer. Moderately priced Yamaha keyboards generally respond to the controls above **Left/Right** but not to the lower controls. Experiment with your own keyboard to see which parameters are recognized.

All sliders have the range 00h (0) to 7Fh (127). A value of -1 appears if you move the slider all the way to the left. This condition designates that a MIDI message for that control will not be included in a saved file. The controls have the following functions:

- **Volume.** The default volume of the voice. Use this control to raise or lower an individual voice relative to the others.
- **Reverb/depth.** Effect of the performance space – a higher value gives a longer echo effect.
- **Chorus.** Simulation of multiple players (like a string section). With a high value, the synthesizer broadens the frequency width of overtones. A low value corresponds to a solo instrument.
- **Cutoff/brightness.** A low-pass filter – a high value passes all frequencies while a low value passes only low frequencies for a mellow sound.
- **Resonance/timbre.** Instruments create a series of overtones, integer multiples of the fundamental note frequency. A instrument like a banjo has a high content of overtones, while a flute has a low content. A high value of this parameter gives strong overtones, while a low value gives reduced overtones.
- **Attack time.** Instruments like pianos typically do not produce notes with uniform amplitude. Instead, the amplitude follows an envelope like that of Fig. 3: a louder section when the hammer strikes the string, a sustained period while the key is held down and then a decay when the key is released. The profile of Fig. 3 is called an *ADSR* envelope (attack/decay/sustain/release). Generally, the amplitudes are built into the parametric definitions of the synthesizer, but you can adjust the timings. Use this control to adjust the attack time. Note that there are only A, D and R commands – the S time is controlled by how long you hold the key.
- **Release time.** The release time of the *ADSR* envelope.
- **Left/right.** This command, also know as *Pan*, controls stereo effects. For a setting of 00h, the voice sounds only in the left channel. A setting of 7Fh gives sound in the right channel.

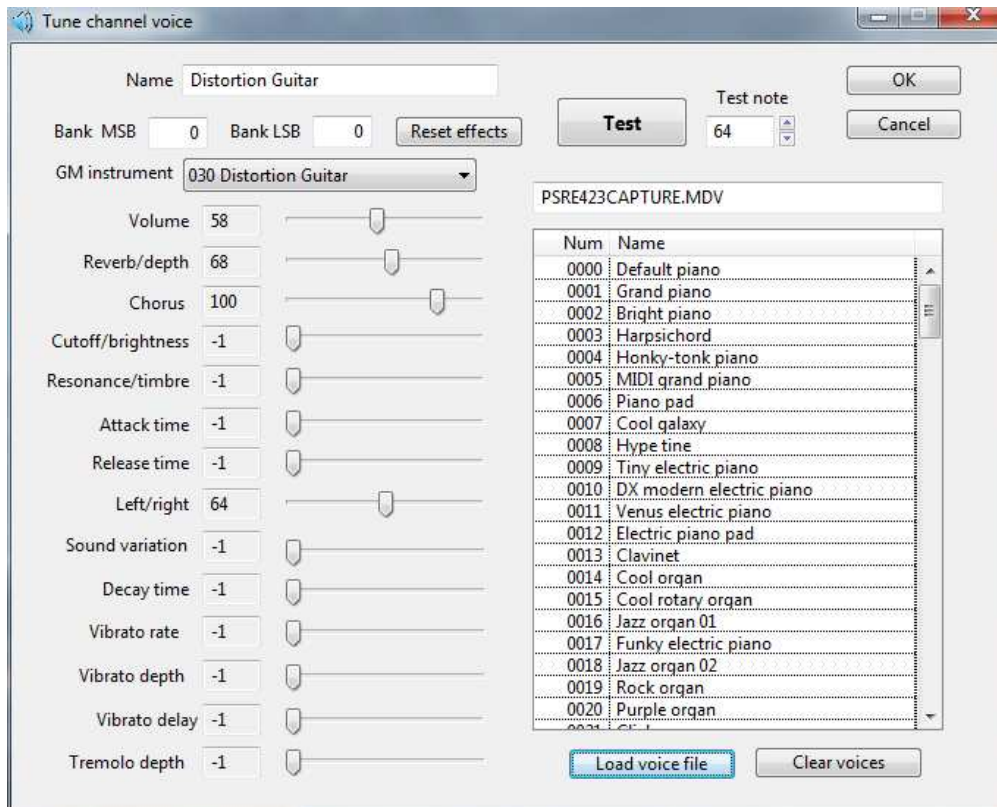


Figure 2: Dialog to adjust channel voice settings.

- **Sound variation.** A generalized control for synthesizers that support multiple variants of an instrument sound.
- **Decay time.** The decay time of the *ADSR* envelope.
- **Vibrato rate.** Vibrato is a low-frequency modulation of pitch. This control sets the modulation frequency.
- **Vibrato depth.** This control sets the frequency range for the vibrato modulation, generally small compared to the fundamental frequency.
- **Vibrato delay** This control sets the phase of the vibrato modulation relative to the *NoteOn* signal.
- **Tremolo depth** Tremolo is a low-frequency modulation of amplitude. This control sets the volume change.

MIDI Doctor updates the output synthesizer as you make changes. Click the **Test** button to hear the current voice. Use the up-down arrows to change the test note to hear the voice in different ranges³. When you are satisfied, click **OK** to update the voice and exit the dialog. Use the play controls to hear how the modified voice sounds in the file, with or without the other channels. Use the **Undo** button if you want to restore the previous voice setting.

³If you are editing a percussion channel, the value of the **Test** note determines the type of percussion sound.

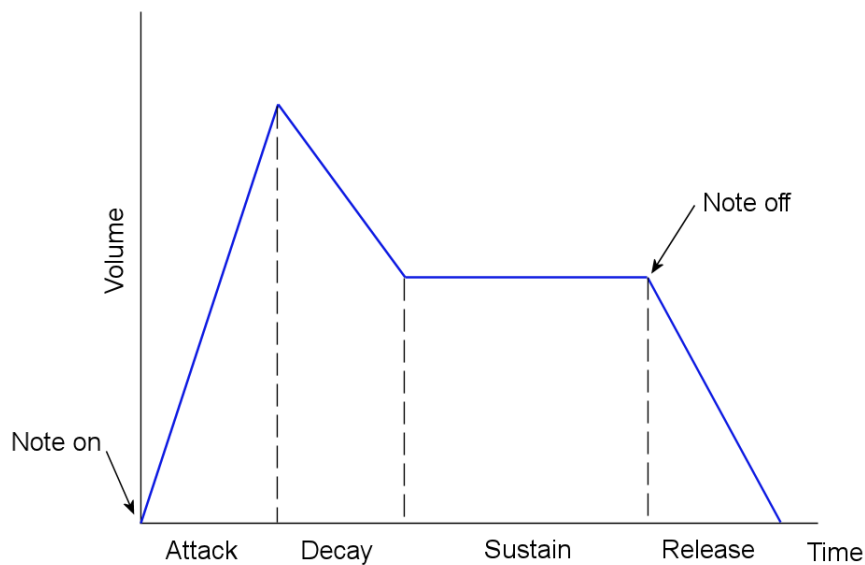


Figure 3: Parameters for note amplitude envelope.

MIDI voice files (MDV) contain control and program messages that represent the instrument collections installed on digital keyboards. In other words, a voice file consist of a large set of presets optimized for particular keyboards. You can save time and effort by using presets. Use the **Load voice file** button and enter the directory **Voices** where we have included data for 47 keyboards prepared from INST specifications. When the file is loaded, double-click on a row to transfer the information to the controls on the left-hand side. You can make changes before saving the voice.

7 Writing the output file

Use the **Save** button (F5) to write an output MIDI file in the Type 1 format described in Sect. 3. Only information from channels set to **ON** will be included. You can refine the type of information saved with the **Output settings** controls:

- **Starting tempo.** On load, **MIDI Doctor** sets this value from the first tempo message in the file. Change the number if you want the output file to play at a slower or faster speed. The program computes an adjustment factor (the ratio of the output to the input tempo) and adjusts all tempo messages accordingly. The effect is to speed up or slow down the entire song. Because of roundoff errors, the output tempo may vary by ± 1 qpm from the designated value.
- **Relative volume.** The slider determines a relative adjustment factor for the volume byte in all **NoteOn** messages: 0.5 (full left), 1.0 (middle) and 2.0 (full right). The factor works in conjunction with the channel volume settings in the voice editor window (Sect. 6). Note that amplification may cause saturation if the calculated value of the volume byte exceeds 7Fh (127).
- **Transpose.** Shift the note values of **NoteOn** and **NoteOff** messages up or down by a specified number of chromatic steps. The effect is to change the range and key signature of the piece. If the input MIDI file contains a key signature message, the value is displayed on load in the **Key signature out** box. The key signature changes as you adjust the **Transpose** value. Unfortunately, the quality of MIDI files you find on the Internet varies considerably, and some may not contain a key signature message. In this case, you may need to use a MIDI editor to determine the input key and to calculate the transposition necessary for a desired output key.
- **Tempo changes (after start).** Sometimes, MIDI files may contain a large number of tempo messages because the transcriber was either particularly expressive or did not use a metronome. Such tempo changes are undesirable if you are preparing an accompaniment. If you uncheck the box, all tempo messages after the first one will not be included in the file.
- **Pitch wheel messages.** Pitch wheel messages shift the frequency of the synthesizer to give a twangy or bluesy sound. Uncheck this box if you want all notes to sound at their prescribed pitch.
- **Program/control messages.** Program messages set the **GM** numbers of channels. Voice control messages (discussed in Sect. 6). These messages may cause a conflict if you are working with a digital workstation with a virtual instrument setup. Uncheck this box to exclude all program messages and voice control messages.
- **Include XG settings.** Unchecking this box excludes all **MSB** and **LSB** control messages from the output stream. Use this option if you want to ensure compatibility with all **GM** compliant devices.

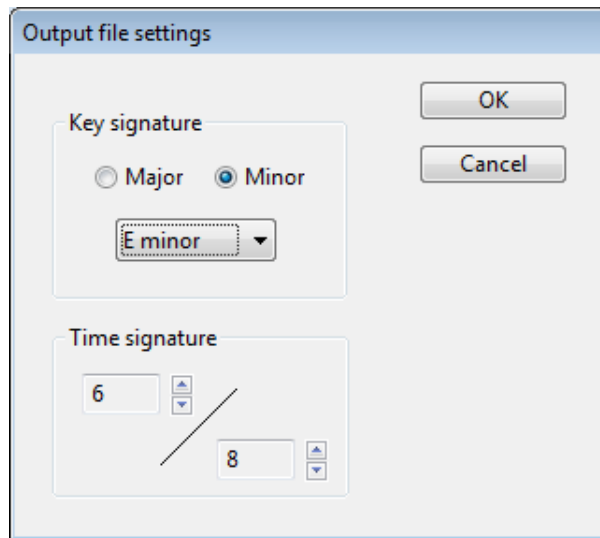


Figure 4: File settings dialog.

- **Text messages and lyrics.** The words in karaoke files are sometimes stored as text messages and sometimes as lyric messages. Uncheck these boxes if you want to remove the information, converting a KAR file to a standard MID file. This option is useful for creating scores with music notation programs. Some programs attempt to include lyrics, making a messy display.
- **System exclusive messages.** These messages are extended structures that contain binary instructions for specific hardware devices. They are generally ignored, so you can remove them from files intended for general distribution.
- **Miscellaneous messages.** Specialized non-MIDI messages that are usually not required to play the file, including SMPTE, MIDI port, MIDI channel, copyright, cue point, instrument and sequence track name.

Finally, sometimes input MIDI files may lack messages to define the key signature and time signature, or the information may be incorrect. In this case, you can set values to record in the output file. Click the **Advanced** button or right-click in any open area of the program window to display the popup menu. Choose **File settings**. The dialog of Fig. 4 is opened. To set the key, first set the **Major/Minor** radio buttons and then choose a key from the popup menu below. Use the up/down arrows to set the time signature. Click **OK** to save the changes and return to the program. Note that these settings do not affect the musical content of the MIDI file.

8 The MIDI voice file specification

This section contains advanced information for developers that is not essential to use **MIDI Doctor**. The MIDI voice-file is a compact entity for saving detailed information on sets of MIDI voice specifications (MIDI patches). Voice files are used in the **Accompaniment Machine** and other KBD-Infinity software to define melody voices to be used with styles for song performances.

MIDI voice files have names of the form `FilePrefix.MDV`. Following the lead of style files, voice files are standard Type 0 (single track) MIDI files that can be read by any MIDI device. The format is compact (the full set of voice specifications for a keyboard may occupy 10 kB) and simple. It is relatively easy to create software routines to transfer voice information to any MIDI device.

Figure 5 shows a screen display of **MIDI Microscope** with a voice file loaded. As with all MIDI files, the first section is the *Header Chunk*:

1. UInt8: 4Dh 54h 68h 64h to represent the characters **MThd**.
2. UInt32: The number 6, always the length of data in the header chunk.
3. UInt16: The number 0, the MIDI file type
4. UInt16: The number 1, the number of tracks
5. UInt16: Pulses per quarter note, set to the arbitrary value of 100 (64)

There is a single track section that begins with

1. UInt8: 4Dh 54h 72h 6Bh to represent the characters *MTrk*.
2. UInt32: The number of bytes in the track.

The data that follow consists of a set of standard MIDI messages (with no *System Exclusive* messages). Each message is preceded by a timing byte, always 00h. The first message must be a non-MIDI type text message that contains the name of the first voice:

```
FF 01 StringLength VoiceNameString
```

The quantity *StringLength* is the length of the following string expressed as a variable-length quantity. The next *StringLength* bytes are ASCII values to define *VoiceNameString*. The voice designator is followed by any number of voice definition messages that a software program ports directly to the MIDI device. All messages are three bytes in length except *Program change* (2 bytes). All parameter values are in the range 00h → 7Fh. All messages are directed to MIDI channel 1 (00). The software is responsible for redirecting the information to other channels for multiple voices. Not all commands may be recognized by all MIDI devices. Unrecognized commands are ignored. The following messages are allowed:

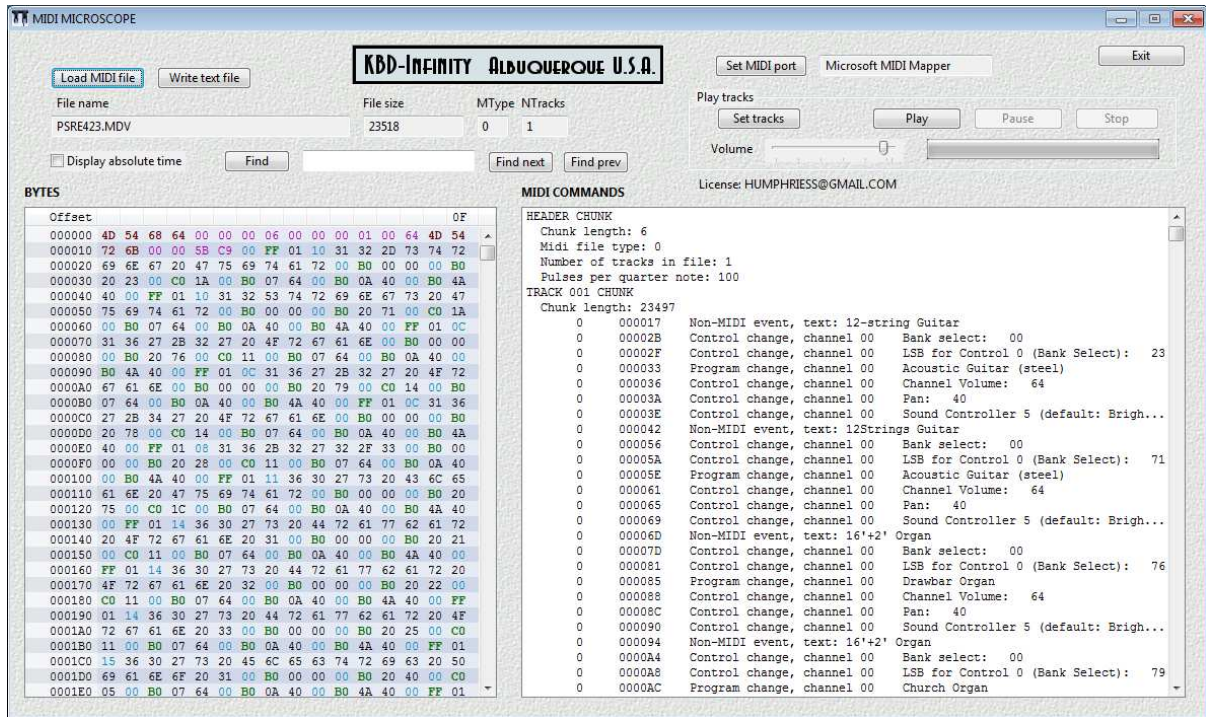


Figure 5: Screenshot of Midi Microscope with a voice file loaded.

XG parameters:

B0 00h BankMSB

B0 20h BankLSB

General MIDI instrument (program change): C0h GMInst

Channel volume: B0h 07h ChanVolume

Pan (left/right speakers): B0h 0Ah Pan

Sound variation: B0h 46h SoundVariation

Timbre (resonance): B0h 47h Timbre

Release time: B0h 48h ReleaseTime

Attack time: B0h 49h AttackTime

Brightness (cutoff): B0h 4Ah Brightness

Decay time: B0h 4Bh DecayTime

Vibrato rate: B0h 4Ch VibratoRate

Vibrato depth: B0h 4Dh VibratoDepth

Vibrato delay: B0h 4Eh VibratoDelay

Depth (reverb): B0h 5Bh Depth

Tremolo depth: B0h 5Ch TremoloDepth

Chorus: B0h 5Dh Chorus

There is an additional non-MIDI message that does not directly affect the output MIDI device, but may be used by the software. The octave displacement parameter is defined by the proprietary message

FF F7 01 (OctaveDisp + 64)

The value 01 is the length of the data (one byte). For an octave displacement of -1, the software should add or subtract $12 * OctaveDisp$ from note values in *NoteOn* and *NoteOff* messages.

The end of data for one voice is signaled by the text message header for the next voice:

```
FF 01 StringLength NextVoiceNameString
```

The end of data for the last voice is designated by a end-of-track message:

```
FF 2F 00
```