

The KMid Handbook

**Antonio Larrosa Jiménez
Pedro Lopez-Cabanillas**



The KMid Handbook

Contents

1	Introduction	6
1.1	KMid's Features	6
2	General Usage	7
2.1	Using MIDI	7
2.1.1	External Devices	7
2.1.2	Sound Cards with a Hardware Synthesizer	7
2.1.3	Software Synthesizers	7
2.2	Opening Songs	8
2.3	Playing Songs	8
2.4	Displaying Lyrics	8
3	Configuration Dialog	9
3.1	Lyrics Settings	9
3.2	MIDI Settings	9
3.3	Soft-Synths Settings	9
3.3.1	FluidSynth	9
3.3.2	TiMidity++	10
4	Playlists	11
4.1	Creating a Playlist	11
4.2	Adding Songs to a Playlist	11
4.3	Removing Songs from Playlists	11
4.4	Playing Order	11
4.5	Selecting a Song from a Playlist	12
4.6	Deleting a Playlist	12
5	MIDI Maps	13
5.1	What is a MIDI Map ?	13
5.2	Do I Need a MIDI Map ?	13
5.3	Creating a MIDI Map	13
5.3.1	The PATCHMAP Section	14
5.3.2	The KEYMAP Section	14
5.3.3	The CHANNELMAP Section	15
5.3.4	The OPTIONS Section	15
5.4	Using MIDI Maps	15

The KMid Handbook

6	Advanced Features	16
6.1	Song Settings	16
6.2	The Channel View	16
6.3	The Piano Player view	16
6.4	The Rhythm Controls	17
7	Key Bindings	18
8	Frequently Asked Questions (FAQ)	19
9	Final Notes	20
9.1	Some Tips and Tricks	20
9.2	Hall of KMid's Friends	20
9.3	About the Authors	21
9.4	Copyrights and License	21
A	Installation	22
A.1	How to Obtain KMid	22
A.2	Requirements	22
A.3	Compilation and Installation	22

Abstract

KMid is a MIDI/karaoke multimedia player

Chapter 1

Introduction

KMid is KDE's MIDI and karaoke multimedia player. It offers some features not found in any other UNIX[®] MIDI player, such as real-time graphics and karaoke text highlighting among others.

KMid currently runs on Linux[®] only, but support for other operating systems is planned, and hopefully will be available in the future. It uses the ALSA sound driver, so it should run on every system where KDE and ALSA can compile.

KMid shows the lyrics in the main window, highlighting the words with a different color while the music is playing, so it is very easy to follow the songs.

1.1 KMid's Features

These are some of KMid's main features:

- It has a very *friendly user interface* to display karaoke text with *real-time highlighting* of lyrics.
- It features a graphical view of what is being played on each MIDI channel, by highlighting the keys pressed in (virtual) piano keyboards.
- The most powerful *Midi Mapper* that you will ever find in any operating system.
- *Drag & drop* so you can drop in KMid any MIDI file from a file manager like Konqueror or Dolphin.
- You can *change the tempo* of songs to play them slower or faster at your wish.
- It shows a visual metronome to easily follow the rhythm of the song.
- *Customizable* font and color for karaoke text.
- Supports the two standards used to include lyrics in MIDI files: lyrics events and text events (guessing which one a song uses automatically).
- Session Management. If a song is playing while you logout from KDE, the next time you login, the same song will start playing.
- *Adjustable volume and pitch* in real-time.
- It can play broken MIDI files which make other players core dump.
- Low CPU usage, depending on the complexity of the song.
- Supports external MIDI synthesizers and software synths.
- Runs on Linux[®] with ALSA support.

Chapter 2

General Usage

2.1 Using MIDI

KMid outputs MIDI data, so it needs a MIDI receiver to render audible sound. There are several kind of MIDI receivers, or Synthesizers, that you can use. The only common required feature is ALSA MIDI support.

2.1.1 External Devices

These are peripheral devices, connected to the computer using a MIDI cable. Modern devices may also include an USB interface. The external MIDI devices are electronic musical instruments, that usually can be played in standalone mode without requiring a connected computer.

To connect a MIDI musical instrument to the computer, you need a MIDI interface in the computer side and a MIDI cable. Some sound cards include MIDI sockets, or cables. There are also cheap USB MIDI interfaces. ALSA MIDI drivers are usually available for them, and absolutely required for KMid usage.

If your MIDI musical instrument does not understand General MIDI standard messages, you may need a MIDI map. See the [MIDI Maps chapter](#) for more information.

2.1.2 Sound Cards with a Hardware Synthesizer

These sound cards were very popular in the past, but much less common nowadays. Some examples are the classic GUS and AWE sound cards, or the SoundBlaster Audigy. All of them are wave-table based synths. Even older are the FM synths, like the Ad Lib. Anyway, they can render MIDI into audio in real-time, with the requirement of ALSA MIDI drivers installed. For wave-table based synths, it is also necessary a collection of sound samples, in the format provided by the sound card manufacturer.

2.1.3 Software Synthesizers

Since more RAM memory and CPU power is available in modern computers, the number and usage of software synths is growing quickly. It is also a cheap alternative, as the only required hardware is a simple sound card, with only audio support. The integrated audio device in many computer motherboards is adequate in most cases.

Two examples of applications capable of rendering MIDI data into audio streams are TiMidity++ and FluidSynth. Both have ALSA sequencer support, and are wave-table based synthesizers.

There are options in the [Configuration dialog](#) to launch these soft synth programs automatically at startup.

To use TiMidity++ you should install a GUS patches collection, like freepats, or a SoundFont (.SF2) file. You can start the program in daemon mode, for instance using the timidity-daemon package in Debian systems.

To use FluidSynth you should install a SoundFont (.SF2) file. It may be a good idea to install also the QSynth GUI front-end. Two excellent and free sound fonts for FluidSynth are [General User GS](#), and Fluid R3, included in some Linux[®] distributions.

2.2 Opening Songs

You can open a song in several different ways.

First, you can select **Open...** from the **File** menu, then you are presented with a standard open dialog, with which you can select the song you wish to open.

You can drag a file from Konqueror or Dolphin and drop it into the KMid window. You can also Drag & Drop multiple songs at the same time.

You can specify one or more song file names in the command line when running KMid. URL arguments are supported, so you can click and open a web link or a file name in your file manager.

You can select a song from the list of recently played files, that are displayed by the menu option **File** → **Open Recent**.

And the final way is by selecting the song from the list of songs in the active playlist.

Anyway, the saved song settings from the last time you played the song are automatically retrieved, if the option **Song** → **Song Settings** → **Automatic** is enabled.

2.3 Playing Songs

To play a song, first open it, and then press on the **Play** button of the toolbar, choose the **Play** entry of the **Song** menu, or just press the playback shortcut, by default the **Media Play** key.

Note that when you open a file, KMid will start playing it automatically. This behavior can be disabled using the **Auto-Start** option in the **Song** menu. If you drag and drop more than one file, they will be added to a playlist and they will be played sequentially.

Once KMid is playing a song, you can move the time slider, by pressing with the middle mouse button mouse button, to go to a specified position.

If a song is playing too fast or too slow for you, you can use the tempo slider. To get back to the default tempo, just press the "100%" button.

2.4 Displaying Lyrics

There are two methods to store lyrics in a MIDI song file: by using **Text events** or **Lyrics events**. Some songs use the first method, some the second, both of them, or none at all when the song does not include lyrics. KMid automatically selects the most appropriate method for each song.

The SMF standard specification allows only ASCII text to be embedded inside a MIDI file. However, you can find many MIDI files with texts encoded in different ways. The text encoding of the lyrics can be selected using a combo box located in the main window. The selected encoding will remain until some other encoding is chosen, and will be remembered for the next session.

Chapter 3

Configuration Dialog

To open the **Configure** dialog, you can use the menu **Settings** → **Configure KMid...**, or the corresponding toolbar icon.

The settings take effect as usual when you press the **OK** or **Apply** buttons.

3.1 Lyrics Settings

- **Font face.** Here you can set the typography used to show the song lyrics in the main window.
- **Font color.** This is the color used for the song text when it has been already played.
- **Font size.** Here you can set the font size for the song lyrics in the main window.

3.2 MIDI Settings

- **Output MIDI Connection.** This list shows the available MIDI connections detected in your system, as reported by the native MIDI subsystem of your operating system. You should choose one of them, that will be connected to the program's MIDI output port.
- **Show advanced MIDI ports.** The advanced MIDI ports are seldom used MIDI devices that are filtered out from the list of available MIDI connections unless you select this checkbox.
- **MIDI mapper file.** Here you can choose the MIDI mapper file that will be used with your external MIDI synthesizer. You can find more information about MIDI mappers in the [MIDI Maps chapter](#).
- **Reset Mode.** This MIDI message is optionally sent to the connected MIDI output port before starting to play each song, resetting several parameters that could be changed by the last song file. Here you can choose a standard message type recognized by your synthesizer, or a custom system exclusive file.

3.3 Soft-Synths Settings

3.3.1 FluidSynth

- **Run FluidSynth at startup.** If you select this checkbox, the FluidSynth soft synth will be automatically launched at startup.

- **Command.** Here you must set the path and name of the soft synth command. Below this text box there is a label reporting the detected program version, and an icon indicating if the minimal requirements to use this program are met.
- **Audio Output.** You should choose here the audio system that will be used by the soft synth program.
- **Audio Device.** Here you can specify an audio device name for the soft synth output. It is usually safe to leave this setting blank.
- **Sample Rate.** Here you can specify the audio sampling rate used by the soft synth to generate digital audio. It is usually safe to leave this setting blank.
- **Sound Font.** Here you must set the path and file name of the SoundFont (.SF2) used by the soft synth. This argument is mandatory.
- **Arguments.** Here you can specify other optional arguments that may be used with your soft synth. It is usual to leave it blank.

3.3.2 TiMidity++

- **Run TiMidity++ at startup.** If you select this checkbox, the TiMidity++ soft synth will be automatically launched at startup.
- **Command.** Here you must set the path and name of the soft synth command. Below this text box there is a label reporting the detected program version, and an icon indicating if the minimal requirements to use this program are met.
- **Audio Output.** You should choose here the audio system that will be used by the soft synth program.
- **Audio Device.** Here you can specify an audio device name for the soft synth output. It is usually safe to leave this setting blank.
- **Sample Rate.** Here you can specify the audio sampling rate used by the soft synth to generate digital audio. It is usually safe to leave this setting blank.
- **Arguments.** Here you can specify other optional arguments that may be used with your soft synth. It is usual to leave it blank.

Chapter 4

Playlists

A playlist is a list of MIDI files that you put in a set, and which are played one after another. This section will help you to use them, and will give you some useful tips to make a good use of them.

4.1 Creating a Playlist

To create a playlist, first open the **Manage Playlist** dialog, by selecting the **Edit Playlist...** entry of the **Playlist** menu, and add your files to the playlist. Finally, select the **Save Playlist...** entry of the **Playlist** menu, and provide the folder and file name for your playlist.

The playlist file is a simple text file that you can edit, copy and move using your usual file management tools.

4.2 Adding Songs to a Playlist

How to Use **Auto-Add to Playlist** Menu Option

If **Auto-Add to Playlist** is enabled, when you open a song (using **File** → **Open...** or Drag & Drop) it (they) will be added to the active playlist without user intervention.

If **Auto-Add to Playlist** is not enabled, when you open a song a temporary playlist will be created and cleared, and the opened songs will be added to it.

4.3 Removing Songs from Playlists

To delete a song from a playlist, just open the **Manage Playlist** dialog, select the song you wish to delete, and then click on the **Remove** button, pressing finally the **OK** button to close the dialog window.

4.4 Playing Order

The songs are played in the same order they are set in the playlist. You can edit the order of the songs using the **Manage Playlist** dialog.

The **Repeat** mode loops the playlist, repeating the first song again when the last song has finished.

When you select the **Shuffle** mode, KMid will randomize the order of the songs in the playlist. The playlist will be shuffled each time you add a new file in the active playlist, and when you activate the **Shuffle** mode.

4.5 Selecting a Song from a Playlist

You can select a song among the songs of the playlist using the **Manage Playlist** dialog.

You can also change to the next song by using the **Next Song** entry of the **Song** menu, the **Next Song** button of the toolbar, or pressing the user configured shortcut key.

To change to the previous song, use the **Previous Song** entry of the **Song** menu, the **Previous Song** button of the toolbar, or press the user configured shortcut key.

4.6 Deleting a Playlist

To delete the contents of a playlist, simply open the **Manage Playlist** dialog and click the **Reset** button.

To delete a playlist file stored on disk, use your favorite file management tool.

Chapter 5

MIDI Maps

5.1 What is a MIDI Map ?

A MIDI map is something that maps MIDI events in other MIDI events.

This is totally needed if a synthesizer does not understand the standard events (that is, if a synthesizer is not General MIDI compliant), in this case, a MIDI map will translate General MIDI events in the events that the synthesizer understands.

For example you can make a MIDI map that converts all the `Change patch to Bright Piano` events, to `Change patch to Trumpet` events, and so when a song tries to play a piano, it will play a trumpet instead.

This may sound odd, (why playing a trumpet when the song is made to play a piano?), but it is very useful. The GM standard specifies that when a MIDI keyboard receives a patch change event to 0, it will change the current patch to `Grand Piano`, but an older synthesizer will change for example to a `Electric Guitar` when it receives a 0. This old keyboard, needed to receive a 3 (for example) to change to a `Piano`. And here comes the MIDI map in action, changing all `change patch to 0` to `change patch to 3` and thus really playing the correct instrument when it has to.

5.2 Do I Need a MIDI Map ?

In short, if you do not have an external synth, the answer is *no*.

If you only have a sound card with MIDI synth, MIDI maps are not likely needed because almost all the sound cards are GM compliant (this include AWE cards, GUS cards, FM devices and so on).

If you are playing music through an external synthesizer, and it is not GM compliant, you will need to make a MIDI map for your MIDI device. Although you will perhaps spend a whole afternoon doing your map file and trying different values for all the options, you will be fully rewarded when you finish it, because you will be able to correctly listen many song files that would sound wrong otherwise.

5.3 Creating a MIDI Map

There is not any program to generate MIDI maps, so you will have to edit a file by hand (using your favorite text editor).

The KMid Handbook

A Midi map is a text file that keeps all the needed translations there will be made when playing music.

It consist of four sections: `PATCHMAP`, `KEYMAP`, `CHANNELMAP` and `OPTIONS`. Each section must appear only once.

The general structure of a map file is an ini style file:

```
[PATCHMAP]
...
# comment line

[KEYMAP]
...

[CHANNELMAP]
...

[OPTIONS]
...
```

Each one of the four section names between square brackets starts the corresponding section. You can put comments by starting the line with a # character. Each section is a collection of several lines of key=value pairs.

Please, do not forget to send me your map file by email, so that future releases of KMid will include support for more non General Midi compliant keyboards.

5.3.1 The `PATCHMAP` Section

This section is used to specify how patches are going to be mapped, from GM to your keyboard specs. The general usage is:

```
(Name of GM Patch name)=(N)
```

Where *N* is the number that you keyboard needs to receive to change the patch to the same that the GM standard does .

Please note that the left side of the equal sign is a symbolic key, so you can change the order of the lines and omit any of the 128 instruments.

5.3.2 The `KEYMAP` Section

The `KEYMAP` section is used to specify how percussion keys are going to be mapped. The usage is:

```
[KEYMAP]
C 0 =0
C#0 =1
D 0 =2
...
```

As with the `PATCHMAP` section, you can change the order of the lines, and omit any of them. Only the first four characters of each key are taken into account.

Keymaps are only used to map keys in the percussion channel. Have a look at the distributed maps to see some examples.

5.3.3 The CHANNELMAP Section

This section can be used to map some channels to different ones. For example, if you want to swap the first and second channels, you can easily do it within the CHANNELMAP section.

However it is more useful for keyboards that need the percussion channel to be in a given channel (the GM standard use the channel 10, others use the channel 16 and others use channel 9).

Note that MIDI devices use 16 channels, so the CHANNELMAP section, has 16 lines, from 0 to 15 , as this one:

```
(N) = (M)
```

Where N is the channel which is mapped to the M channel.

5.3.4 The OPTIONS Section

The OPTIONS section has some general options that can be very useful:

```
[OPTIONS]
PitchBenderRatio = r
MapExpressionToVolumeEvents = true
ForceDrumsPatch = 30
```

You can specify all options, only one, or none of them.

The PitchBenderRatio r value, has the ratio by which pitch bender events will be multiplied. That is, when a MIDI file tries to send a pitch bender event with a n value, the real value that will be sent is $n * (r/4096)$ (the 4096 value is for not having to put decimal points in the map file).

This is used because the GM standard says that when a MIDI keyboard receives a Pitch Bender event with a 4096 data value, it should bend the note to a higher one , but some MIDI keyboards try to bend the initial note by two or more higher notes (even an octave higher!) when they receive a 4096. This can be easily fixed by trying different values so that instead of sending a 4096, KMid sends the appropriate value.

When the MapExpressionToVolumeEvents option is set in the map file, and a MIDI file try to send an expression event, KMid will send a volume event which is understood by more non-GM keyboards, and which has a similar effect. There are many MIDI files which use expression events to fade out at the end of a song, so if you feel that music should be heard softer and softer, you can turn on this option and see if this is what you need, because your MIDI synthesizer could be ignoring the expression events because it does not understand them.

The ForceDrumsPatch option may be useful to always use the percussion patch on the percussion channel.

5.4 Using MIDI Maps

To use a MIDI map, open the **Configuration** dialog by selecting the **Configure KMid...** entry of the **Settings** menu.

Then click the **MIDI** category icon, and select the map file in the **MIDI mapper file** combo box.

Chapter 6

Advanced Features

6.1 Song Settings

For each played song, several options can be saved and automatically retrieved the next time the song is opened. These options are:

- Text encoding
- Volume
- Pitch (tone, transposition)
- Rhythm tempo
- Channel labels
- Channel buttons
- Channel instruments

If you do not want automatic saving and retrieving, you can use the **Load** and **Save** menu options available in the menu **Song** → **Song Settings**.

6.2 The Channel View

The Channel view is a table of 16 rows, one for each MIDI channel. In each row there is a text box where you can label the channels for your own reference, buttons to enable the **Solo** and **Mute** functions, a vu-meter, and also a **Lock** button and a combo box where you can see and change the musical instrument for each channel.

Only the active channels (the channels used by the current song) have enabled controls.

6.3 The Piano Player view

The Piano Player view can show up to 16 virtual keyboards, one for each MIDI channel. Only the active channels (the channels used by the current song) can be enabled. The virtual keyboards not only show the played notes in real-time, but can be used also to trigger notes using the mouse or the computer keyboard.

6.4 The Rhythm Controls

The rhythm controls are: a tempo label, showing the song tempo in beats per minute, a visual metronome including a measure/part counter, and a tempo slider to modify the song tempo, that is, the speed of the song.

If you changed the tempo, you can get back the original one clicking on the **100%** button.

Chapter 7

Key Bindings

Key	Action
Media Play	Play the loaded song.
Media Stop	Stop playing
Ctrl+Q	Quit KMid
F1	Open this document

Chapter 8

Frequently Asked Questions (FAQ)

1. *What is exactly a MIDI file?*

A Midi file contains information about how to play a song, that is, it contains the notes, the rhythm, velocity, etc. This implies that the same MIDI file, when played in two different devices can produce very different results. It is like a musical score, that can be performed very differently by two different musicians.

2. *I can get better sound with a mp3/wav player, why should I use KMid?*

The size of a typical MIDI file is about 50 KB. while the typical mp3 filesize is 4 MB. And with a good synthesizer device, you can get a comparable sound quality. Even more, with a MIDI file, you can change individual instruments, change the velocity of a song, etc. so you have more overall control.

Another point of view about this: comparing MIDI and mp3 is like comparing painting with photography. Or comparing SVG with jpeg. Each one has different properties and use cases.

3. *I cannot get my sound card to work with KMid, what can I do?*

First, you should verify your ALSA installation and configuration. Second, you need to connect KMid to a MIDI synthesizer. If your sound card does not have one, use a software synth instead.

4. *I want to add a whole folder to a playlist, but having to add the MIDI files one by one is not funny.*

I agree, that's why KMid supports Drag & Drop. Just open, in Konqueror or Dolphin, the folder you want to add, select all the files, drag them and drop them in KMid.

Be sure to set the **Auto-Add to Playlist** option before, so that the files will be added to the current playlist. If you do not do this, files will be added to a new temporary playlist.

5. *I cannot follow the lyrics, it's playing too fast!*

You can use the rhythm control tempo slider to make it play slower. Remember that you can press the **100%** button to restore the original tempo.

Chapter 9

Final Notes

9.1 Some Tips and Tricks

I will include some tips so that you can take fully advantage from all the features of KMid:

Opening files

I always keep a KDE desktop with a Konqueror or Dolphin window in my root MIDI folder, and KMid in this desktop (if playing a MIDI file) or sticky (if playing a karaoke file :-)). This way, when the active playlist finishes, or I want to play some file, I just go to the Konqueror or Dolphin window, select the desired files and Drag & Drop to the KMid's window.

Suppose that you want to play some MIDI files, but do not want to add them to any playlist, well, just turn off the **Auto-Add to Playlist** option in the **Playlist** menu, and open the files, they will be added to a temporary playlist.

Another method to create a new playlist

Suppose that you have MIDI files A.mid, B.mid and C.mid. At first you only want to play the first MIDI file, so you unselect **Auto-Add** and open A.mid. You get then a Temporary playlist with only one MIDI file.

Then you decide to play also B and C, and make a playlist with all them, what do you do?

Easy, select **Auto-Add** and open B.mid and C.mid (by any of the multiple methods), they will be automatically added to the temporary playlist, that will then have A.mid, B.mid and C.mid. At this point, you can use the **Save Playlist** dialog, enter the name of the new playlist, and you are done. You already have a new playlist, which holds the A, B and C MIDI files, and that is not deleted when you close KMid.

9.2 Hall of KMid's Friends

These are some folks who have sent MIDI files or a postcard, thanks to everyone!

- Ola Sigurdson - Taking Care of Business (Bachman Turner Overdrive)
- EG Lim - A really very nice postcard from Penang.
- Guenther Starnberger - Amadeus (Falco) and Schrei Nach Liebe (Die Aerzte)
- Leandro Terrés - All That She Wants and The Sign (Ace of Base)
- Nick Stoic - Two MIDI files

9.3 About the Authors

KMid was originally made by Antonio Larrosa Jiménez, from Málaga (Spain). It has been rewritten, currently maintained and published with the name KMid by Pedro Lopez-Cabanillas, from Spain as well.

Thanks to the following persons their help in developing KMid:

- Paul J. Leonard P.J.Leonard@bath.ac.uk - Support for AWE cards
- Sebestyen Zoltan szoli@digoo.inf.elte.hu- FreeBSD port and AWE testing
- Christian Esken esken@kde.org - For organizing the KDE multimedia efforts
- Stephan Kulow coolo@kde.org- Configure scripts and help with **automake** and **CVS**
- Jaroslav Kysela - Help in doing the Linux[®] Ultrasound Project driver support
- Takashi Iwai and Joseph H. Buehler - Fix for AWE cards pitch being too high
- Adrian Knoth - For giving me good news and many suggestions
- Kevin Street - Patch to support FreeBSD 3.0
- Thanks go also to Jose Luis Sanchez for his testing of GUS support, Ignacio Garcia for testing the AWE support, Hans Petter Bieker, Ola Sigurdson, Marc Diefenbruch, Peter Gritsch, Magnus Pfeffer, Urko Lusa, Peter-Paul Witta, Thorsten Westheider, Ulrich Cordes and everyone that sent me a patch, bug report or just an email to give me encouragement.
- And of course to all the fabulous musicians over the net that keep giving us those wonderful MIDI and karaoke files.

9.4 Copyrights and License

Program and documentation copyright 1999-2001 Antonio Larrosa Jiménez

Program and documentation copyright 2009-2010 Pedro Lopez-Cabanillas

This documentation is licensed under the terms of the [GNU Free Documentation License](#).

This program is licensed under the terms of the [GNU General Public License](#).

Appendix A

Installation

A.1 How to Obtain KMid

KMid can be found on its homepage: <http://sourceforge.net/projects/kmid2>. In the homepage, you can follow the development, see some information about it, some screenshots, etc.

A.2 Requirements

KMid requires to work:

- KDE Platform
- Linux[®] with ALSA sound, MIDI drivers, and the ALSA library.
- A sound card. A good soundcard and/or external synthesizer are recommended, as the sound quality depends greatly in your soundcard.

A.3 Compilation and Installation

For detailed information on how to compile and install KDE applications see [Building and Running KDE Software From Source](#)

Since KDE uses **cmake** you should have no trouble compiling it. Should you run into problems please report them to the KDE mailing lists.

There are some examples installed in `$KDEDIR /share/apps/kmid`