

DOSBox

Install DOSBox Staging for a more enhanced DOS gaming experience:

```
yay -S dosbox-staging
```

General MIDI/Soundfonts

The integrated FluidSynth MIDI sequencer has issues with some soundfont files, resulting in minor to major music playback issues in games. Timidity++ does not have these issues.

To install simply:

```
pacman -S timidity++
```

A list of available soundfonts to install from the AUR, [sorted by votes](#):

AUR Package Name	Description
<code>soundfonts-aur-meta</code>	Installs all the soundfont packages in the AUR
<code>soundfont-unison</code>	A lean and clean GM/GS soundbank
<code>soundfont-sgm</code>	A balanced, good quality GM soundbank
<code>soundfont-titanic</code>	A public domain, high quality MIDI soundfont by Luke Sena
<code>soundfont-generaluser</code>	A small and well balanced GM/GS soundbank for many styles of music.
<code>soundfont-zeldamcsf2</code>	Legend of Zelda: Minish Cap soundfont for MIDI playback
<code>soundfont-zelda3sf2</code>	Legend of Zelda: Link to the Past soundfont for MIDI playback
<code>soundfont-fatboy</code>	A free GM/GS SoundFont for classic video game MIDI, emulation, and general usage
<code>soundfont-arachno</code>	GM/GS soundbank courtesy of Maxime Abbey.
<code>soundfont-sso-sf2</code>	The Sonatina Symphonic Orchestra by Mattias Westlund. (SF2 format)
<code>soundfont-toh</code>	Don Allen's Timbres of Heaven soundfont

AUR Package Name	Description
<code>soundfont-opl3-fm-128m</code>	A SoundFont designed to simulate the classic MIDI sound of the Sound Blaster 16 (and other YM262 enabled hardware).
<code>soundfont-sunshine-perc</code>	Five drum/percussion soundfonts from Sunshine Studios. Non-commercial use only.
<code>soundfont-realfont</code>	GM soundbank by Michel Villeneuve
<code>soundfont-personalcopy</code>	A large free SoundFont.
<code>soundfont-jeux</code>	Jeux organ soundfont

Configure Timidity++ to use the soundfont of your choosing in its global config file

`/etc/timidity++/timidity.cfg`:

```
soundfont /usr/share/soundfonts/FluidR3_GM.sf2
```

Set up `timidity++` to run in daemon mode and start with user login:

```
systemctl --user enable --now timidity
```

You need to tell DOSBox which MIDI Port to send MIDI data to. Install the `alsa-utils` package and list the available MIDI ports with `aconect`:

```
pacman -S alsa-utils
aconect -o
```

The output might look something like this:

```
client 14: 'Midi Through' [type=Kernel]
  0 'Midi Through Port-0'
client 128: 'TiMidity' [type=User,pid=89573]
  0 'TiMidity port 0 '
  1 'TiMidity port 1 '
  2 'TiMidity port 2 '
  3 'TiMidity port 3 '
```

In the configuration file for DOSBox, pass the client ID of the sequencer and the port to use on the `midiconfig` setting. The mididevice needs to be `default`. The syntax is `[client]:[port]`:

```
[midi]
mpu401      = intelligent
mididevice = default
```

```
midiconfig = 128:0
```

Gravis UltraSound (GUS)

The Gravis UltraSound cards were technically advanced soundcards with sample-based music synthesis ("wavetable") and hardware-mixing. DOSBox can emulate a Gravis UltraSound card for games that support it.

To enable GUS emulation, set the following options in your DOSBox configuration file:

IMPORTANT: The `ultradir` references a directory **within** DOSBox, not your local filesystem!

```
gus      = true
gusbases = 240
gusirq   = 5
gusdma   = 3
ultradir = C:\ULTRASND
```

Depending on where you mount your `C:` drive (e.g. `~/DOS`), the `ULTRASND` directory needs to be placed inside it.

Installing GUS drivers

NOTE: Assumptions being made in this guide:

- The `C:` drive is mounted from `~/DOS`
- The `X:` drive is mounted from `~/Downloads/GUS Install` and contains the GUS setup files

IMPORTANT: Make sure you turn on GUS emulation in DOSBox **before** starting the setup procedure!

Preparations

GUS emulation needs the original install disks for the Gravis UltraSound, which can be downloaded [here](#).

Create an `autoexec.bat` at the root of DOSBox's `C:` drive:

```
touch ~/DOS/autoexec.bat
```

In DOSBox

Extract the contents into a directory and mount it as drive `X:` in DOSBox:

```
mount x ~/Downloads/GUS Install
```

Change directory to the `GUS410` directory and start the installer:

```
X:  
cd X:\GUS410  
INSTALL.EXE
```

Setup procedure:

1. Choose `Restore`, **NOT** `Install`
2. When asked what to restore, provide the glob pattern `*.*`
3. Keep the default target drive letter
4. Keep the default target directory
5. Start the installation process

Back at the main menu:

1. Choose `Install` (since it is restored, the installation should be quick)
2. Keep the defaults for the drive and directory
 - If it can't find Windows, provide `C:\ULTRASND\WINDOWS`
3. When the installation completes successfully exit out
 - Don't run Express or Custom Setup

Change directory to the `GUS411` directory and start the installer:

```
cd X:\GUS411  
INSTALL.EXE
```

Repeat the installation steps above.

Testing

To test if setup was successful restart DOSBox, change into `C:\ULTRASND` and start `MIDIDEMO.BAT`.

If you hear music being played, the installation was successful.

Games with CD Audio

You can use [CBAE](#) to save some space with games that use CD audio tracks by compressing them.

`cbae` is a NodeJS package that is installed via `npm`. If you don't have NodeJS already installed:

```
pacman -S nodejs
```

Then install the `cbae` package **globally**:

```
npm i cbae --location=global
```

`cbae` takes `.bin/.cue` images as input and uses the information of the `.cue` file to determine what the CD audio tracks are.

To convert a `.bin/.cue` image:

```
cbae e KEEPER.cue -o ./ -enc OPUS:64 -p $(nproc)
```

This achieves the following:

- `e KEEPER.cue`: encodes CD audio tracks of the image `KEEPER.cue`
- `-o ./`: outputs the resulting files into a sub-directory of the current directory, e.g.:

```
KEEPER.bin
KEEPER.cue      <-- input file
KEEPER [e]     <-- sub-directory
├─ KEEPER.cue          <-- new .cue file by cbae
├─ KEEPER - Track 01.bin <-- binary game data
├─ KEEPER - Track 02.opus <-- CD audio track
├─ KEEPER - Track 03.opus
├─ KEEPER - Track 04.opus
├─ KEEPER - Track 05.opus
├─ KEEPER - Track 06.opus
└─ KEEPER - Track 07.opus
```

- `-enc OPUS:64`: encodes audio tracks with Opus at 64 kbps (see `cbae --help` for available codecs)
- `-p $(nproc)`: specifies how many CPU cores are used for encoding (`$(nproc)` assigns the maximum number of cores available)

Mount the newly created `.cue` file with DOSBox's `imgmount` command, e.g. as the D: drive:

```
imgmount d ~/DOSGAMES/KEEPER [e]/KEEPER.cue -t cdrom
```

Revision #2

Created 2023-03-26 01:56:04 UTC by Sebin

Updated 2023-08-08 22:44:35 UTC by Sebin