

GeneralUser GS Documentation

GeneralUser GS: **version 2.0.1** (10/15/2024)

Documentation: **revision 3** (10/15/2024)

by **S. Christian Collins**

Updates to this documentation will be published on [my GeneralUser GS web page](#) and viewable online at [GitHub](#).

Table of Contents

1. About GeneralUser GS	
2. Compatibility	
2.1. Operating Systems	
3. Configuration and Usage	
3.0.1. BASSMIDI	
3.0.2. FluidSynth	
3.1. DAW plugins (Composing or Live Musical Performance)	
3.1.1. FluidSynth Plugin (GNU/Linux, macOS, Windows)	
3.1.2. BassMidi VSTi (Windows)	
3.1.3. juicysfplugin (macOS, Windows)	
3.1.4. LMMS (GNU/Linux, macOS, Windows)	
3.1.5. Others	
3.2. Playing MIDI Files	
3.2.1. FluidSynth (GNU/Linux, macOS, Windows)	
3.2.2. Falcosoft Soundfont Midi Player (Windows)	
3.2.3. KQ Sampei (iOS)	
3.2.4. SpessaSynth (Web App)	
3.2.5. Midi Clef Karaoke Player (Android)	
3.2.6. dmidiplayer (GNU/Linux, macOS, Windows)	
3.2.7. Others	
3.3. Retro PC Gaming	
3.3.1 DOSBox Staging / DOSBox-X	
3.3.2 SCUMMVM	
3.3.3. Virtual MIDI Devices	
3.4. All-Purpose MIDI Playback (virtual MIDI device + separate MIDI player)	

3.4.1. Qsynth (GNU/Linux)	
3.4.2. CoolSoft VirtualMidiSynth (Windows)
3.4.3. OmniMIDI (Windows)
4. Preset List
4.1. Instrument Presets
4.2. Percussion Presets
5. Support Files
6. Compressing GeneralUser GS for Web and Mobile Apps
7. Website and Contact

1. About GeneralUser GS

[GeneralUser GS](#) is a [Roland GS](#) and [General MIDI \(GM\)](#) compatible SoundFont bank for composing, playing MIDI files, and retro gaming. It features 261 instrument presets and 13 drum kits, all while possessing a very low memory footprint (30.7 MB of RAM). GeneralUser GS features detailed instrument programming, making heavy use of SoundFont synthesis and modulator features. This can help it sound as good or better than SoundFonts 2–3 times its size but also makes GeneralUser GS more dependent on having a standards-compliant synth for proper playback of its instruments.

I decided on the name “GeneralUser” because the bank was designed to sound well with all kinds of music rather than specializing in only a few styles, thereby benefiting the general user. It’s not a great name, but that’s what I came up with as a teenager in the late ‘90s, and I’ve kept it ever since.

2. Compatibility

GeneralUser GS is fully compatible with [FluidSynth](#) and [BASSMIDI](#) (version 2.4.14.28 or later) and most software based on these two engines. It is also fully compatible with [KQ Sampei](#) (iOS) & [SpessaSynth](#) (web app).

Most other SoundFont synths will not properly render many of the presets and should be avoided. The following table shows the compatibility status of several popular SoundFont engines:

SoundFont Synth / Application	Compatibility	Notes
Apple AUMIDISynth / DLSMusicDevice, et al. (most MIDI players on macOS/iOS)	Awful	The Apple SoundFont synths are an abomination, violating the spec in every way imaginable.
BASSMIDI (& OmniMIDI, et al.)	Excellent	BASSMIDI v2.4.14.28 or later <i>required</i> . Configuration may be necessary (see section 3.0.1).
bismark bs-16i (Android, iOS)	Poor	bs-16i cannot handle GeneralUser's modulators, so many instruments sound incorrect.
FluidSynth (& Qsynth, et al.)	Excellent	FluidSynth v2.3 or later recommended. See section 3.0.2 for ideal settings.
KQ Sampei	Excellent	Version 1.6 or later recommended. See section 3.2.3 for ideal settings.
MuseScore Studio 4	Poor	The current version of MuseScore disables the lowpass filter, breaking many presets.
Sobanth VSTi	Good	Uses the wrong mod envelope attack curve, so some presets will sound incorrect.
Sound Blaster Audigy / X-Fi family of sound cards	Okay	Many instruments will sound too bright due to Creative Labs' filter implementation.
Sound Blaster Live! family of sound cards	Poor	The Live! only supports SoundFont v2.0, but v2.01 or later is required for GeneralUser GS.
SpessaSynth	Excellent	Version v3.20.43 or later recommended.

SoundFont Synth / Application	Compatibility	Notes
SynthFont , et al.	Poor	Can't handle GeneralUser's modulators, making quite a mess of many presets.
TinySoundFont	Poor	Currently out of spec in crucial ways, no modulator support.

If you would like to test a SoundFont player for proper [specification](#) support, you can use my [SoundFont Spec Implementation Test](#).

2.1. Operating Systems

Windows and Linux currently have the best application support for high quality SoundFont playback. Support on macOS is weak in comparison, with many applications opting to use macOS' horrible native SoundFont player. As for mobile operating systems, Android has a few good options, but software for iOS is not covered in this documentation as I do not own any contemporary iOS devices for testing.

As for Linux, all of the Windows applications that I list in the following sections also work quite well in [Wine](#). I personally run the Windows VST plugins in [REAPER](#) via [yabridge](#) with great success.

3. Configuration and Usage

The following sections will detail how to best configure your SoundFont synth for use with GeneralUser GS. The sections are broken down into the most common use cases for GeneralUser GS: [using DAW plugins](#) (composing or live musical performance), [playing MIDI files](#), and [retro PC gaming](#). There are also options that can cover [multiple use cases](#). Most of the recommended applications will be using either BASSMIDI or FluidSynth and will be listed in order of my personal recommendation.

I won't detail every application using these two SoundFont engines, but instead only what I find to be the easiest to use and/or provide the best playback quality. There are also likely to be good applications that I have missed, so please [let me know](#) if you think a different application should be on this list.

Before I detail per-app configuration, here are some general notes and settings for BASSMIDI and FluidSynth:

3.0.1. BASSMIDI

Notable features of BASSMIDI include:

- Low CPU usage, ideal for playing black MIDI files.
- Supports additional controllers from the GS and GM2 standards that are not officially supported in the SoundFont standard.
- Doesn't implement the entire SoundFont spec (not all modulators are supported), but covers the most important features.

GeneralUser GS requires BASSMIDI version 2.4.14.28 or later. BASSMIDI versions earlier than 2.4.14.33 will also need to have the `BASS_MIDI_FONT_NOLIMITS` flag enabled to avoid emulating legacy Sound Blaster hardware limitations. Beginning with BASSMIDI 2.4.14.33, Sound Blaster emulation is disabled by default.

If the above requirements aren't met, many instruments will sound too bright and some will behave incorrectly. Applications based on BASSMIDI may come bundled with an older version of BASSMIDI, in which case you can usually download an updated version of the BASSMIDI engine to replace it with. This is easy to do, and instructions for each recommended application are included in the sections below.

You can download the latest release of BASSMIDI by scrolling down to the "Add-ons" section of [the BASS audio library homepage](#). As of the writing of this documentation, the latest release version is 2.4.14.1. If this is still the case, you will need to instead download the beta version linked on [this forum post](#). The downloaded zip file will contain two copies of `bassmidi.dll`. The copy inside the `x64` folder is the 64-bit version, and the copy outside of that folder is the 32-bit version.

A further improvement to the sound can be achieved by adjusting the reverb and chorus settings as follows, where permitted:

- Reverb level: **60**
- Chorus type: **Chorus 3**
- Chorus level **43**

This is, of course, a matter of preference. If you want the true Roland/Yamaha General MIDI experience, then you can leave the reverb and chorus levels at 64, but I find the default effects levels to be a bit overbearing.

3.0.2. FluidSynth

Notable features of FluidSynth include:

- A fully open source, complete implementation of the SoundFont specification.
- The most accurate SoundFont rendering available, including full modulator support.
- Excellent-sounding reverb and chorus engine (when configured correctly).

FluidSynth v2.3 or later is recommended for use with GeneralUser GS, but earlier versions should sound pretty good as well. For best sound quality, set FluidSynth's sample interpolation method to "7th order" (the default is "4th order"). If CPU usage is a concern, then leave it at "4th order".

The default reverb and chorus settings in FluidSynth v2.3.x and earlier are not very good (chorus is too aggressive, reverb sounds like a small room). FluidSynth v2.4 [will fix this](#) by providing much nicer defaults. Fortunately, these new effects settings can be used in earlier versions of FluidSynth in programs that allow configuration of the reverb and chorus engines.

Here is a table comparing the default reverb and chorus settings between FluidSynth 2.3.x, 2.4, and my recommended settings for use with GeneralUser GS:

Parameter	v2.3.x	v2.4	Recommended
synth.reverb.damp	0	0.3	0.3
synth.reverb.level	0.9	0.7	0.7
synth.reverb.room-size	0.2	0.5	0.5
synth.reverb.width	0.5	0.8	0.8
synth.chorus.depth	8	4.25	3.6
synth.chorus.level	2	0.6	0.55
synth.chorus.nr (voices)	3	3	4
synth.chorus.speed (Hz)	0.3	0.2	0.36

Instructions for setting these values can be found in the individual app sections below.

3.1. DAW plugins (Composing or Live Musical Performance)

If you wish to use GeneralUser GS for composing music in a digital audio workstation (DAW), there are some very nice virtual instrument plugins available that feature excellent SoundFont compatibility and high quality audio rendering. I have listed my favorite plugins below in order of preference. Where I have not provided installation instructions, please follow each plugin's documentation for installation and loading onto a track in your DAW.

3.1.1. FluidSynth Plugin (GNU/Linux, macOS, Windows)

Some may be turned off by [FluidSynth Plugin](#)'s lack of a traditional GUI, but there are some really nice features here. This plugin provides the best SoundFont renderer available (FluidSynth), and allows full customization of FluidSynth's options including enabling/disabling/customizing the reverb and chorus, changing the sample interpolation method, loading only the active preset's samples to RAM, and more. These settings are written as text into the interface, so you will want to consult the plugin's documentation to see all of the options.

Installation & Setup

1. [Download the latest FluidSynth Plugin release](#) for your platform.
2. Extract the downloaded archive, then locate the `FluidSynthPlugin.vst3` folder, which will be inside `programs/VST3`. For macOS, you may wish to use the Audio Units (AU) plugin instead of the VST3 to provide a better experience on Apple Silicon Macs. The AU version is the `FluidSynthPlugin.component` folder, which will be inside `programs/AU`.
3. Copy the `FluidSynthPlugin.vst3` folder to your platform's default VST3 location:
 - **Linux:** `~/.vst3`
 - **macOS:** `/Library/Audio/Plug-ins/VST3`
 - **Windows:** `C:\Program Files\Common Files\VST3`

For the AU plugin on macOS, copy the `FluidSynthPlugin.component` folder into `/Library/Audio/Plug-ins/Components`.

4. *Windows only:* Recent Windows versions of FluidSynth (v2.3.5 and earlier) have a [broken chorus effect](#). This has been fixed in FluidSynth v2.3.6. The current version of FluidSynth Plugin (v0.7.1) comes bundled with FluidSynth v2.3.5, so you will need to manually upgrade the FluidSynth library to v2.3.6 if you want the chorus to work properly. Follow these steps:

1. Download the latest Windows release of FluidSynth from [the FluidSynth Github page](#).

2. Open the downloaded .zip file and copy the `/bin/libfluidsynth-3.dll` file into `C:\Program Files\Common Files\VST3\FluidSynthPlugin.vst3\Contents\x86_64-win\`, replacing the copy found there.
5. *Linux only:* Install your system's libfluidsynth package (`libfluidsynth3` in ubuntu, `fluidsynth-libs` in Fedora). Also note that FluidSynth Plugin might not work on older Linux OS versions (requires Ubuntu 23.10 or later; I'm not sure about Fedora or others).
6. Open your DAW and let it scan for new/updated VST plugins. You should now be able to add "FluidSynthPlugin (DrTT)" to a track as a virtual instrument.
7. *macOS only:* There is a good chance macOS will tell you it can't open FluidSynth Plugin since it is from an unverified publisher. Unfortunately, the usual step of allowing the program to run via "System Preferences" → "Security & Privacy" won't work for VSTs. Instead, you will need to open the Terminal app (found in the Utilities folder) and run the following command (assuming you placed the VST/AU folder in the default location in step 3):

For the VST3 plugin:

```
sudo xattr -rd com.apple.quarantine /Library/Audio/Plug-Ins/VST3/FluidSynthPlugin.vst3
```

For the AU plugin:

```
sudo xattr -rd com.apple.quarantine /Library/Audio/Plug-Ins/Components/FluidSynthPlugin.component
```

Then, re-scan the VST plugins in your DAW.

8. After adding "FluidSynthPlugin (DrTT)" to a track in your DAW, right-click in the FluidSynth Plugin GUI and choose "Select SoundFont File...". A file selection window should appear.
9. Find and choose `GeneralUser-GS.sf2` and click "Open".
10. Click "Confirm" to load the SoundFont.

11. Paste the following recommended settings into the plugin's text box:

```
synth.gain = 1.0
synth.interpolation-method = 7
synth.reverb.active = 1
synth.chorus.active = 1

synth.reverb.damp = 0.3
synth.reverb.level = 0.7
synth.reverb.room-size = 0.5
synth.reverb.width = 0.8

synth.chorus.depth = 3.6
synth.chorus.level = 0.55
synth.chorus.nr = 4
synth.chorus.speed = 0.36
```

12. Press "Confirm" again.

13. Right-click in the FluidSynthPlugin GUI and choose "Select Program..." to choose your desired preset. Click "OK" and then "Confirm" to activate the preset.

Alternatively to the last step, you can opt not to choose a preset and instead rely on preset change MIDI messages to do so. Also, if you prefer to use your DAW's effects plugins instead of FluidSynth's, you can set both `synth.reverb.active` and `synth.chorus.active` to `0`, leaving out the rest of the reverb and chorus settings lines. Whatever your preferred settings end up being, you can save the plugin configuration as a preset to avoid having to write the values in every time.

3.1.2. BassMidi VSTi (Windows)

As should be obvious from the plugin name, [BassMidi VSTi](#) is based on the BASSMIDI library. GeneralUser GS requires BassMidi VSTi version 1.6 or later, and "Do not emulate SB HW limits" must be enabled in "Soundfont/Bassmidi Setup". Future plugin versions will likely make the Sound Blaster hardware emulation opt-in rather than opt-out, so be sure to read the wording of the option. Basically, you *don't* want Sound Blaster hardware limit emulation when using GeneralUser GS.

BassMidi VSTi is the only plugin on this list that is multi-timbral, meaning you can have different instruments loaded on each of its 16 MIDI channels.

To use GeneralUser GS with BassMidi VSTi:

1. Click the "Soundfont/Bassmidi Setup" button in the BassMidi VSTi GUI.
2. Under "Default Soundfont", click the "File" button. A file selection window should appear.
3. Find and choose `GeneralUser-GS.sf2` and click "Open".
4. Enable the "Do not emulate SB HW limits" option if using plugin version 1.6. This option will likely not be available or needed in future plugin versions.
5. Set Reverb level to **60**.
6. Set Chorus type to "Chorus 3".
7. Set Chorus level to **44**.
8. Click "OK".

3.1.3. juicysfplugin (macOS, Windows)

If you want to use FluidSynth with an more conventional GUI (compared to FluidSynth Plugin), [juicysfplugin](#) is your best option. However, there are some limitations and quirks you should be aware of:

- MIDI event timing will be limited to your current audio buffer size, with minimum timing resolution = 64 samples. This means if you have your ASIO buffer set to 512 samples, MIDI events can only be registered to juicysfplugin every 512 samples, leading to inaccurate timing. The workaround for rendered audio is to set your rendering buffer size to 64 samples. REAPER has a setting for this, but I'm not sure about other DAWs.
- Reverb and chorus engines are disabled.
- It is not possible to change the sample interpolation method from FluidSynth's default 4th-order resampler. This means there will be cases where high frequency aliasing may be audible, particularly when playing a sample far above or below its natural sample rate.
- The entire SoundFont is always loaded to RAM, even if you're only using one preset from an otherwise enormous SoundFont.

If any of the above are an issue for you, you should consider using FluidSynth Plugin (see [section 3.1.1](#) above) instead.

3.1.4. LMMS (GNU/Linux, macOS, Windows)

GeneralUser GS works great in LMMS' integrated "Sf2 Player", which is based on FluidSynth. If you wish to use Sf2 Player's built-in reverb and chorus, I recommend the following values:

- Reverb
 - size: **0.5**
 - damp: **0.3**
 - width: **0.8**
 - level: **0.7**
- Chorus
 - num: **4**
 - level: **0.55**
 - speed: **0.36**
 - depth: **3.6**

Please note that it doesn't appear possible to control the reverb and chorus levels via MIDI CC91 or CC93, so these effects would only be present on SoundFont presets that call for them. It will therefore usually be preferable to achieve reverb or other effects using LMMS' effects chains.

3.1.5. Others

One other DAW plugin of note is [Sobanth](#) (Windows). This plugin is an almost-perfect implementation of the SoundFont standard, but the current version has a few issues, including:

- The modulation envelope attack phase is linear instead of convex, impacting many synth sounds, solo brass, clarinet, and others.
- Stereo samples are panned incorrectly. This doesn't impact GeneralUser GS, which uses mono samples only.

If an update to Sobanth ever fixes the issues mentioned above, I will provide full configuration instructions within this documentation, as this plugin is otherwise quite good.

3.2. Playing MIDI Files

Playing MIDI files on PC is usually accomplished in one of two ways:

1. The MIDI file player and SoundFont synthesizer are both contained within the same application.
2. The MIDI file player and SoundFont synthesizer are separate applications connected via a virtual MIDI interface.

Standalone applications (*option #1*) are generally the easiest to set up and use, but as of this writing, only a few provide ideal playback quality. This will change in the future once FluidSynth-based MIDI players such as VLC adopt FluidSynth 2.4 with its improved reverb and chorus settings (the current FluidSynth version as of this writing is 2.3.6). Using a separate synthesizer and MIDI player (*option #2*) is covered in [section 3.4 below](#).

The following sections start with my top recommended MIDI playback applications for each operating system, followed by other noteworthy options. I have included a few MIDI files with GeneralUser GS in the `demo MIDI's` folder for testing each application. You will also find .ogg audio of each MIDI file as a reference for how playback should sound (played using FluidSynth and GeneralUser GS 2.0.0).

3.2.1. FluidSynth (GNU/Linux, macOS, Windows)

[FluidSynth](#) can usually be found embedded within other applications (such as VLC) for MIDI playback, but it can also be run as a standalone MIDI player and/or virtual MIDI device. I find FluidSynth's combination of sound and MIDI playback quality to be the best of all the options I have tried, provided you make a few tweaks.

The player runs in a terminal window, but you can easily set it up to automatically play when double-clicking a MIDI file in your file manager. If you'd rather use something with a graphical user interface (GUI), check out the other players listed in the sections below.

Installation & Setup



Note: I have created video tutorials of the setup process for each operating system:

[GNU/Linux](#) | [macOS](#) | [Windows](#)


1. [Download](#) and install FluidSynth for your platform:
 - **Linux:** A `fluidsynth` package should be available in your distro's repository.
 - **macOS:** Install using [Homebrew](#) (`brew install fluidsynth`) or [MacPorts](#) (`sudo port install fluidsynth`).

- **Windows:** For a proper installation, I recommended installing FluidSynth via [Chocolatey](#) (`choco install fluidsynth`). Otherwise, you can get a zip file of the [latest release here](#), but you will need to manually extract it and move the `bin` , `include` and `lib` folders to your desired application folder (e.g., `C:\Program Files\FluidSynth`).
2. Open an editor that can create plain text files. Most operating systems come with one. You will use this the editor to create a FluidSynth configuration file:
- **Linux:** The text editor included will vary between distros. Common text editors on Linux include *GEdit* (GNOME), *Kate* or *Kwrite* (KDE Plasma).
 - **macOS:** *TextEdit* comes preinstalled on macOS. After opening TextEdit, go to "Format" → "Make Plain Text" to create the correct file type.
 - **Windows:** *Windows Notepad* comes preinstalled on Windows.
3. Paste the following into the editor, adjusting the path to `GeneralUser-GS.sf2` to match its location on your system:


```
set synth.default-soundfont "/path/to/GeneralUser-GS.sf2"
set synth.device-id 16
set synth.polyphony 512
set synth.gain 0.5
set synth.reverb.damp 0.3
set synth.reverb.level 0.7
set synth.reverb.room-size 0.5
set synth.reverb.width 0.8
set synth.chorus.depth 3.6
set synth.chorus.level 0.55
set synth.chorus.nr 4
set synth.chorus.speed 0.36
interp 7
```

This will tell FluidSynth to function as a Roland GS device (`set synth.device-id 16`), set the reverb and chorus engines to sound much better than default, set sample interpolation to the highest quality (`interp 7`), and point to GeneralUser GS. If you need to adjust audio driver or other settings, more options are listed in [the FluidSynth documentation](#).

4. Save the configuration file, placing and naming it as is expected for your OS:
- Linux, macOS: `$HOME/.fluidsynth`
 - Windows: `%USERPROFILE%\fluidsynth.cfg`

 **Note:** Be sure that your text editor doesn't add a ".txt" extension at the end of the filename when saving. On Windows `%USERPROFILE%` points to your Windows User folder, usually located at `C:\Users\YourUserName\`. On macOS, `$HOME` points to `/Users/YourUserName/`, and on Linux, `$HOME` usually points to `/home/YourUserName/`. Obviously, replace "YourUserName" with your user name.

You can now play MIDI files using the command line (via your system's terminal application), e.g., by typing `fluidsynth midifile.mid`. To make this process easier, you can type `fluidsynth` into the terminal (note the space after "fluidsynth"), and then drag and drop your MIDI file from your file browser into the terminal window. This will add the MIDI file location to the command line, and then just hit the Enter key to start playing. Type `quit` then press Enter to stop playback / quit FluidSynth.

 **Note:** By default on Windows, if you didn't install FluidSynth via Chocolatey, the above command won't work unless you provide the full path name for the FluidSynth executable, e.g.: `"C:\Program Files\fluidsynth\bin\fluidsynth.exe" midifile.mid`

Once you have everything working and are happy with the results, you can set FluidSynth as the default MIDI player (it will open when you double-click on a MIDI file). To do this, follow the instructions below for your OS:

- **Linux:**

1. Copy the `fluidsynth.desktop` file from the `support/FluidSynth/Linux/` folder included with GeneralUser GS to the `~/.local/share/applications/` folder. Your OS should now give you the option to choose "FluidSynth" when opening a MIDI file, and you can set it as the default application if desired.

- **macOS:**

1. Using Finder, copy the `FluidSynth` app from the `support/FluidSynth/macOS` folder included with GeneralUser GS into your `Applications` folder.
2. In Finder, right-click (or Control-click) a MIDI file and select "Open With" → "Other...".
3. Choose "FluidSynth" from the applications list.
4. Upon first launching the app, you will be prompted to give FluidSynth access to control the Terminal app. Click "OK".
5. The MIDI file should begin playing in a terminal window. You might need to wait a few seconds for playback to begin, depending on the MIDI file.
6. Once you have verified that MIDI playback is working, you can close the terminal window to stop playback.

7. Next, you will probably want to set FluidSynth to automatically launch when double-clicking on a MIDI file. To do so, right-click (or Control-click) on a MIDI file in Finder, and choose "Get Info".
8. In the Info window that pops up, click the arrow next to "Open with" to expand it, then click the application list drop-down, and choose "Other...".
9. Choose "FluidSynth" from the applications list, and click "Add".
10. Click the "Change All..." button to set FluidSynth as the default player for all MIDI files. You will get a dialog confirming this change (click "Continue").
11. Close the Info window.

- **Windows:**

1. Right-click on a MIDI file and choose "Open with" → "Choose another app".
2. When asked "How do you want to open this file?", select "More apps", then scroll down the list and choose "Look for another app on this PC".
3. Locate `fluidsynth.exe` (note that Windows will hide the ".exe" extension by default). If you installed FluidSynth via Chocolatey, the file will be in `C:\tools\fluidsynth\bin`. If you downloaded the zip file instead, it will be in the `bin` folder wherever you extracted the zip file.
4. Click "Open" to choose the file.

Now, when you double-click a MIDI file, FluidSynth should begin playing it in a terminal. Note that there will be no playback progress indicator. Many MIDI files have a measure or two of silence at the beginning, so give it a few seconds for playback to start. Closing the terminal window will stop playback and exit FluidSynth.

3.2.2. Falcosoft Soundfont Midi Player (Windows)

Based on BASSMIDI, Falcosoft's [Soundfont Midi Player](#) is my #1 recommendation on Windows if you are looking for a MIDI player with a full graphical user experience, assuming you don't mind some manual setup. The version available as of this writing (6.4) will require an update to the BASSMIDI library for correct playback with GeneralUser GS.

Installation & Setup



Note: I have created a [tutorial video for the setup process](#).

First, install Soundfont Midi Player and check to see what version of BASSMIDI it is using:

1. Download [Soundfont Midi Player](#), ideally the 64-bit version (the rest of this tutorial will assume you have downloaded the 64-bit version).

2. There is no installer for this application, so you will need to copy/extract the `MidiPlayer6` folder from inside the downloaded zip file to a location of your choosing (such as your desktop). You will run the MIDI player directly from this folder.
3. Open the extracted `MidiPlayer6` folder and double-click `MidiPlayer.exe` to launch the player.
4. Click the menu button (icon with three lines at the top-left of the window) and choose "About...". In the window that pops up you can find the BASSMIDI version number.

If the BASSMIDI version is lower than 2.4.14.33, you will need to update it manually as follows:

5. Close Soundfont Midi Player.
6. Download the latest BASSMIDI version by following the instructions in [section 3.0.1](#) above.
7. Extract/open the downloaded archive and locate `bassmidi.dll` inside the `x64` folder (unless you installed the default, 32-bit version of Soundfont Midi Player, then locate the copy *outside* of the `x64` folder).
8. Copy that `bassmidi.dll` into the the Soundfont Midi Player folder (`MidiPlayer6`), replacing the existing `bassmidi.dll` in that folder.
9. Repeat steps 3 and 4 above to verify that BASSMIDI is now at version 2.4.14.33 or later.

Then, start Soundfont Midi Player and configure it as follows:

10. Click the gear icon near the top left of the application. This will take you to a "Midi Player Device Settings" window.
11. Under "Midi Out" at the top, enable "Use Bass (Soundfonts/VSTi)". This will expand the window to show "BassMidi Settings".
12. Under BassMidi Settings, click the "File" button under "Default SoundFont", find and select `GeneralUser-GS.sf2` , then click "Open".
13. For improved sound quality, enable "Use Sinc Interpolation". If your CPU usage during MIDI playback ends up being too high, you can revert this change.
14. On the left of the settings window, under "SysEx Effects (Midi Out/BassMidi/VSTi)", set the following:
 - Reverb level: **59**
 - Chorus level: **43**
15. Click OK to close the window.

You can now load and play MIDI files.

3.2.3. KQ Sampei (iOS)

[KQ Sampei](#) is a fantastic SoundFont MIDI player for iOS that can also be used as a realtime virtual instrument. Its new SoundFont engine V3 is a complete implementation of the SoundFont specification and is on par with FluidSynth, in my opinion. KQ Sampei version 1.6 or later is recommended for use with GeneralUser GS 2.0, along with the following app settings:

1. Import `GeneralUser-GS.sf2` as a sound set in "Preferences" → "Sound Set Manager".
2. De-activate the default `GeneralUser GS v1.471.sf3` sound set (tap the red minus sign, then "Unload"), and activate `GeneralUser-GS.sf2` (tap the green plus sign).
3. Tap on `GeneralUser-GS.sf2` to edit the SoundFont's settings.
4. Under "Sampei Engine", make sure "V3" is selected.
5. Turn off "GS Custom Controllers".
6. Exit the SoundFont customization and sound set screens.
7. Click the preferences icon at the top of the app (looks like a gear).
8. Scroll down and set the following options:
 - Slow LPF: **off**
 - Smooth Volume Level: **off**
 - Polyphony: for playing MIDI files, I recommend setting this to **256**
9. Close the "Preferences" popup, and click on "Param" to view the parameter options.
10. Set the reverb level to **37.01%** and then enable "Lock Parameters" just above that.

3.2.4. SpessaSynth (Web App)

[SpessaSynth](#) is a web MIDI player that works great with GeneralUser GS v2.0. In fact, it currently uses GeneralUser GS by default, though the version included with the web player uses compressed audio samples (SF3 format). If you wish to have the best audio quality, you can manually upload the uncompressed SF2 file into the player.

3.2.5. Midi Clef Karaoke Player (Android)

[Midi Clef Karaoke Player](#) (current version is 7.4.8) is the best MIDI player I was able to find for Android. It uses the BASSMIDI engine, though I couldn't find which version. I can at least tell that the included BASSMIDI version is older than 2.4.14.33, so many instruments will sound a bit too bright and some will not quite sound correct (see [section 3.0.1](#) above for an explanation).

You can install Midi Clef Karaoke Player from the Google Play Store, and it shouldn't be too hard to figure out how to load `GeneralUser-GS.sf2` as the default SoundFont and start playing MIDI files. Consult the app's built-in help if you get stuck.

3.2.6. dmidiplayer (GNU/Linux, macOS, Windows)

[Drumstick Multiplatform MIDI File Player \(dmidiplayer\)](#) can play MIDI files using its internal FluidSynth engine or an external MIDI device, among other options. As of version 1.7.4, dmidiplayer contains everything you need for excellent sounding MIDI playback, if you don't mind a few caveats:

1. [MIDI tempo is a bit off in some cases.](#)
2. Slight MIDI event timing jitter even when [setting FluidSynth to use the smallest possible audio period size \(64\)](#). I will need to do further research and update this documentation once I know more.

If you are put off by the idea of using FluidSynth through a terminal, dmidiplayer provides a way to enjoy FluidSynth's sound while using a very nice, simple GUI.

Installation & Setup

1. [Download and install dmidiplayer](#) version 1.7.4 or later.
2. Start dmidiplayer.
3. Go to "Settings" → "MIDI Setup" and set the following:
 - MIDI OUT (box 1): **FluidSynth**
 - MIDI OUT (box 2): **FluidSynth**
4. Click the wrench icon to open the FluidSynth driver settings.
5. Set the appropriate audio driver for your OS. Here are my recommendations:
 - **GNU/Linux:** I recommend using the `alsa` driver, as `pipewire` and `pulseaudio` drivers do not support setting the number of audio periods. If you have your system set up to use Jack for low-latency audio, use the `jack` driver instead.
 - **macOS:** Use the `coreaudio` driver.
 - **Windows:** Use the `dsound` driver. The `waveout` driver performs notably worse than `dsound`, and I haven't been able to get the `wasapi` driver working on any system I have tested.
6. Next, you'll want to configure the FluidSynth audio buffer to reduce MIDI timing jitter as much as possible. Start by setting the period size to **64** and the number of periods to **32**. If you later find audio to be breaking up or contain sonic artifacts, try increasing the number of periods. If you reach the max of **64** periods and find the sound is still problematic, add **64** to the period size, set number of periods to **32** and again gradually increase this number until the sound is clear. If it is still not clear, keep adding **64** to the period size, and repeat the process as needed.
7. Leave the reverb and chorus settings as-is. In case these get changed, you can click the "Restore Defaults" button to reset the values.

8. Next to "Sound Font:", click the "..." button to locate `GeneralUser-GS.sf2` and click "Open".
9. Click "OK"

You should now be able to load and play MIDI files using `dmidiplayer`.

3.2.7. Others

The following applications use FluidSynth for MIDI playback and therefore are supported by GeneralUser GS. However, they do not allow customizing the reverb and chorus effects and will not sound nearly as good as the above recommended options. This will change in the future once these programs adopt the not-yet-released 2.4 version of FluidSynth:

- [Audacious](#) (GNU/Linux, macOS, Windows)
- [VLC Media Player](#) (GNU/Linux, macOS, Windows)

3.3. Retro PC Gaming

GeneralUser GS can be used for music playback in many retro PC games. The following applications all feature FluidSynth for MIDI playback. Please note that I have only tested these programs in Linux, but my instructions should work for all operating systems.

3.3.1 DOSBox Staging / DOSBox-X

[DOSBox Staging](#) and [DOSBox-X](#) are modern forks of DOSBox. While I only use DOSBox Staging, both applications do a fantastic job of playing retro PC games. Whichever you choose, follow the application's documentation for configuring your game(s) to use FluidSynth for MIDI playback, and then apply the following tweaks for the best sound in GeneralUser GS.

DOSBox Staging

1. Locate your DOSBox Staging configuration directory:
 - **Linux:** `~/.config/dosbox/`
 - **Linux Flatpak:** `~/.var/app/io.github.dosbox-staging/config/dosbox/`
 - **macOS:** `~/Library/Preferences/DOSBox/`
 - **Windows:** `C:\Users\<username>\AppData\Local\DOSBox\`
2. *OPTIONAL:* Copy `GeneralUser-GS.sf2` into the `soundfonts` folder within your DOSBox configuration directory ([see instructions here](#)).
3. From your DOSBox Staging configuration directory, open the `dosbox-staging.conf` file in a text editor. All of the parameters are in "key = value" pairs. To change them, find the line

containing the desired key, and then change the value to the right of the equals sign.

4. If you don't like the sound of a compressor adjusting audio levels on the fly, set `compressor = false`.
5. By default, `mididevice =` is set to `auto`, but you can ensure it will use FluidSynth by changing the value to `fluidsynth`.
6. Set FluidSynth to use GeneralUser GS, set the SoundFont attenuation to 55, and customize the reverb and chorus for best sound. Assuming you performed step 2, the FluidSynth configuration should look like this:

```
soundfont      = "GeneralUser-GS.sf2 55"  
fsynth_chorus = 4 0.55 0.36 3.6 sine  
fsynth_reverb  = 0.5 0.3 0.8 0.7  
fsynth_filter  = off
```

If you didn't perform step 2, you will need to provide the full file path to the GeneralUser GS SoundFont.

DOSBox-X

I haven't tested DOSBox-X, but GeneralUser GS should work well with it after configuring the effects.

1. Follow [the documentation] to set up FluidSynth and with GeneralUser GS.
2. Add the following configuration options in the `[MIDI]` section of your DOSBox-X configuration file:

```
fluid.reverb.roomsize = 0.5  
fluid.reverb.damping  = 0.3  
fluid.reverb.width    = 0.8  
fluid.reverb.level    = 0.7  
fluid.chorus.number   = 4  
fluid.chorus.level    = 0.55  
fluid.chorus.speed    = 0.36  
fluid.chorus.depth    = 3.6
```

If you find the game music too loud, try setting `fluid.gain` to `0.5` or lower (the default is `0.6`). [Let me know](#) what value works well in your experience, and I will include it in future versions of this documentation.

3.3.2 SCUMMVM

To set **SCUMMVM** to use GeneralUser GS for General MIDI playback:

1. Go into "Global Options...".
2. Select the "MIDI" tab.
3. Set "GM device:" to "FluidSynth".
4. Click the "SoundFont:" button and browse to select `GeneralUser-GS.sf2` and click "Choose".
5. Click "FluidSynth Settings" and set the following settings:
 - Reverb
 - Active: **Yes**
 - Room: **50**
 - Damp: **30**
 - Width: **80**
 - Level: **70**
 - Chorus
 - Active: **Yes**
 - N: **4**
 - Level: **55**
 - Speed: **36**
 - Depth: **36**
 - Type: **Sine**
 - Misc
 - Interpolation: If you have a fast CPU, set to "Seventh-order" for best sound quality.

3.3.3. Virtual MIDI Devices

If your game or emulator doesn't have a good SoundFont player built-in, you can often set it up to send its MIDI music playback to a virtual MIDI device instead. All of my recommended virtual MIDI devices are covered in the following section.

3.4. All-Purpose MIDI Playback (virtual MIDI device + separate MIDI player)

A virtual MIDI device is a standalone synthesizer application that plays MIDI data received from a separate player entity. The application playing MIDI could be a sequencer, MIDI file player, video game, or just about anything. You will need to first start the virtual MIDI device and then point your MIDI player/game/etc. to output its MIDI to that device. Please follow each program's documentation for instructions on how to do that.

Recommended MIDI File Players

If you wish to play MIDI files, you will need a player that can output MIDI to your virtual MIDI device. If you are using a virtual MIDI device for other purposes (e.g., video games), you can ignore this section. Here are a few standalone MIDI players that I have tried:

- [Drumstick Multiplatform MIDI File Player \(dmidiplayer\)](#) (GNU/Linux, macOS, Windows) – This is a very nice MIDI player with a simple, accessible GUI. I have found Drumstick MIDI Player to occasionally be [slightly off tempo](#) from other players, but you may or may not notice.
- [Ultralight MIDIPlayer](#) (GNU/Linux, macOS, Windows) – This player is written to be lightweight and able to handle highly complex MIDI files. It requires Java 8 to run. Under Linux, I could not get it to see my virtual MIDI devices, so I was unable to see how well it performs on that platform.
- [Sekaiju](#) (Windows) – This player is actually a full-on MIDI sequencer, but it does a great job playing MIDI files to an external MIDI source. It also runs well in Linux via Wine.

There are a lot of MIDI players out there I haven't explored, so I am definitely [open to suggestions](#).

Now, on to the list of recommended virtual MIDI devices.

3.4.1. Qsynth (GNU/Linux)

[Qsynth](#) is a graphical front-end for FluidSynth. While you can use FluidSynth (see [section 3.2.1](#)) directly as a virtual MIDI device, Qsynth makes it a lot easier to configure and tweak FluidSynth's many settings.

Installation & Setup

1. Install Qsynth from your system repository or from Flathub using Flatpak. Here are the Flatpak instructions:

```
flatpak install net.sourceforge.dmidiplayer org.rnbc.b.qsynth
```

2. Start Qsynth and click on "Setup...".
3. On the "MIDI" tab, adjust the following options:
 - MIDI Bank Select mode: **gs**
 - MIDI Client Name ID (ALSA/CoreMidi): **Qsynth1**

4. On the "Audio" tab, I recommend using the `alsa` driver, as `pipewire` and `pulseaudio` drivers do not support setting the number of audio periods. If you have your system set up to use Jack for low-latency audio, use the `jack` driver instead (unavailable in the Flatpak version).
5. Next, you'll want to set the audio buffer to reduce MIDI timing jitter as much as possible. The lower the buffer size, [the tighter FluidSynth's MIDI timing](#). Start by setting the buffer size to **64** and the buffer count to **32**. If you later find audio to be breaking up or contain sonic artifacts, try increasing the buffer count. If you reach the max buffer count of **64** and find the sound to still be problematic, add **64** to the buffer size, set the buffer count to **32** and again gradually increase the buffer count until the sound is clear. If it is still not clear, add **64** again to the buffer size, etc.
6. On the "Soundfonts" tab, click "Open" to find and select `GeneralUser-GS.sf2`, then click "Open", and then "OK" to exit setup.
7. After confirming to restart the FluidSynth engine, adjust the following settings in the main window:
 - Gain: **40**
 - Reverb Room: **50**
 - Reverb Damp: **30**
 - Reverb Width: **80**
 - Reverb Level: **70**
 - Chorus N: **4**
 - Chorus Level: **55**
 - Chorus Speed: **36**
 - Chorus Depth: **36**

OPTIONAL: Set Qsynth to identify as Roland Sound Canvas

I have found that I get more consistently correct MIDI file playback when setting FluidSynth to identify as a Roland Sound Canvas device. This step is optional but I highly recommend it. To adjust the FluidSynth device ID you will need to modify the Qsynth app launcher (`.desktop` file).

If you are using the KDE Plasma desktop:

1. Right-click your panel's application menu icon and choose "Edit Applications..." to bring up the KDE Menu Editor.
2. Locate the entry for Qsynth. On my system, it is in the "Multimedia" category.

3. Add the following at the end of "Command-Line Arguments:" (make sure there is a space between anything that was already there and the following):

```
-o synth.device-id=16
```

4. Click "Save" and exit the KDE Menu Editor.
5. Verify the change by launching Qsynth, then click on "Setup...", go to the "Settings" tab and verify that "synth.device-id" is set to **16**.

If you are using another desktop environment::

1. First, you will need to copy Qsynth's .desktop file into your local applications directory. If you installed the Flatpak version, this can be accomplished via the following command (at least on an Ubuntu-based OS), but you should verify the file paths on your system and adjust as necessary:

```
cp  
/var/lib/flatpak/app/org.rnbc.qsynth/current/active/export/share/applications/org  
.rnbc.qsynth.desktop ~/.local/share/applications/
```

If you installed Qsynth via your distro's `qsynth` package, the `.desktop` file will likely be found in `/usr/share/applications/`. So then you would instead do:

```
cp /usr/share/applications/org.rnbc.qsynth.desktop ~/.local/share/applications/
```

2. Edit `~/.local/share/applications/org.rnbc.qsynth.desktop` in your favorite text editor, and add the following to the end of the `Exec=` line, being sure to add a space between the existing contents of the `Exec=` line and the following:

```
-o synth.device-id=16
```

So, if you are using the Flatpak version, your exec line should look like this:

```
Exec=/usr/bin/flatpak run --branch=stable --arch=x86_64 --command=qsynth  
org.rnbc.qsynth -o synth.device-id=16
```

3. Save the file and exit the text editor.

4. Verify the change by launching Qsynth, then click on "Setup...", go to the "Settings" tab and verify that "synth.device-id" is set to 16.

3.4.2. CoolSoft VirtualMidiSynth (Windows)

CoolSoft VirtualMIDISynth is a virtual MIDI device based on BASSMIDI. As of the current version (2.13.8), it will also need an updated copy of `bassmidi.dll` to sound fully correct with GeneralUser GS, but this is easy to do. Please follow the application's instructions for setup, then make the following tweaks for best results with GeneralUser GS:

1. Download the latest BASSMIDI version by following the instructions in [section 3.0.1](#) above.
2. Copy the 64-bit `bassmidi.dll` into `C:\Program Files\VirtualMIDISynth`, replacing the existing copy.
3. In the VirtualMIDISynth options, adjust the following settings:
 - Use sinc interpolated sample mixing...: **yes**
 - Reverb max level: **60**
 - Chorus max level: **43**

3.4.3. OmniMIDI (Windows)

OmniMIDI is a virtual MIDI device based on BASSMIDI, very similar in function to VirtualMIDISynth. It will also need an updated copy of `bassmidi.dll` to sound fully correct with GeneralUser GS, but this is easy to do. Please follow the application's instructions for setup, then make the following tweaks for best results with GeneralUser GS:

1. Download the latest BASSMIDI version by following the instructions in [section 3.0.1](#) above.
2. Copy the 64-bit `bassmidi.dll` into `C:\Windows\System32\OmniMidi`, replacing the existing copy.
3. Copy the 32-bit `bassmidi.dll` into `C:\Windows\SysWow64\OmniMidi`, replacing the existing copy.

4. Preset List

The following presets are included in GeneralUser GS. Banks are selected using MIDI CCO followed by a preset change (PC) command to select the desired preset. The instrument presets can be accessed on all MIDI channels except for channel 10. The percussion presets can only be accessed on channel 10. I have duplicated the percussion presets to bank 120 within the instrument presets so that the drum kits can be accessed on other channels besides 10.

An instrument list file `GeneralUser GS.ins` is provided in Cakewalk format and can be found in the `support` folder included with GeneralUser GS. This list file is supported by many MIDI applications and will make it much easier to see what preset you are selecting.

4.1. Instrument Presets

<i>Bank:</i>	<i>PC:</i>	<i>Preset:</i>	<i>Bank:</i>	<i>PC:</i>	<i>Preset:</i>
0	0	Grand Piano	0	6	Harpsichord
11		Piano & Str.-Fade	8		Coupled Harpsichord
12		Bell Piano	11		Harpsichord noVel
0	1	Bright Grand Piano	12		Coupled Harpsi noVel
11		Piano & Str.-Sus	0	7	Clavinet
0	2	Electric Grand Piano	0	8	Celeste
0	3	Honky-Tonk Piano	11		Tinkling Bells
0	4	Tine Electric Piano	0	9	Glockenspiel
8		Chorused Tine EP	0	10	Music Box
11		Tine & FM EPs	12		Christmas Bells
12		Bell Tine EP	0	11	Vibraphone
0	5	FM Electric Piano	11		Vibraphone No Trem.
8		Chorused FM EP	0	12	Marimba
11		Piano & FM EP	0	13	Xylophone

<i>Bank:</i>	<i>PC:</i>	<i>Preset:</i>	<i>Bank:</i>	<i>PC:</i>	<i>Preset:</i>
0	14	Tubular Bells	0	24	Nylon Guitar
8		Church Bells	8		Ukulele
9		Carillon	0	25	Steel Guitar
11		Bell Tower	8		12-String Guitar
0	15	Dulcimer	16		Mandolin
0	16	Tonewheel Organ	0	26	Jazz Guitar
8		Detuned Tnwl. Organ	8		Hawaiian Guitar
11		Tonewheel Org noVel	0	27	Clean Guitar
12		Detun Tnwl Org noVel	8		Chorused Clean Gt.
0	17	Percussive Organ	12		Clean Guitar 2
8		Detuned Perc. Organ	0	28	Muted Guitar
11		Percussive Org noVel	8		Funk Guitar
12		Detun Perc Org noVel	0	29	Overdrive Guitar
0	18	Rock Organ	11		Wah Guitar (CC21)
11		Rock Organ noVel	0	30	Distortion Guitar
0	19	Pipe Organ	8		Feedback Guitar
8		Pipe Organ 2	0	31	Guitar Harmonics
11		Pipe Organ noVel	8		Guitar Feedback
12		Pipe Organ 2 noVel	0	32	Acoustic Bass
0	20	Reed Organ	0	33	Finger Bass
11		Reed Organ noVel	0	34	Pick Bass
0	21	Accordion	0	35	Fretless Bass
8		Italian Accordion	0	36	Slap Bass 1
0	22	Harmonica	0	37	Slap Bass 2
0	23	Bandoneon			

<i>Bank:</i>	<i>PC:</i>	<i>Preset:</i>	<i>Bank:</i>	<i>PC:</i>	<i>Preset:</i>
0	38	Synth Bass 1	0	49	Slow Strings
1		Synth Bass 101	1		Slow Strings Mono
8		Acid Bass	11		Velo Strings
11		Techno Bass	12		Velo Strings Mono
12		Mean Saw Bass	0	50	Synth Strings 1
0	39	Synth Bass 2	8		Synth Strings 3
8		Beef FM Bass	11		Synth Strings 4
11		Pulse Bass	0	51	Synth Strings 2
0	40	Violin	11		Synth Strings 5
0	41	Viola	0	52	Concert Choir
0	42	Cello	1		Concert Choir Mono
0	43	Double Bass	0	53	Voice Oohs
0	44	Tremolo Strings	0	54	Synth Voice
1		Tremolo Strings Mono	0	55	Orchestra Hit
0	45	Pizzicato Strings	0	56	Trumpet
0	46	Orchestral Harp	1		Trumpet 2
0	47	Timpani	0	57	Trombone
0	48	Fast Strings	1		Trombone 2
1		Fast Strings Mono	0	58	Tuba
8		Orchestra Pad	0	59	Muted Trumpet
12		Full Orchestra	0	60	French Horns
13		Woodwind Choir	1		Solo French Horn

<i>Bank:</i>	<i>PC:</i>	<i>Preset:</i>	<i>Bank:</i>	<i>PC:</i>	<i>Preset:</i>
0	61	Brass Section	0	78	Whistle
1		Brass Section Mono	11		Whistlin'
8		Brass Section 2	0	79	Ocarina
11		Brass Section 3	0	80	Square Lead
0	62	Synth Brass 1	1		Square Wave
8		Synth Brass 3	8		Sine Wave
0	63	Synth Brass 2	12		Square Lead 2
8		Synth Brass 4	13		Square Lead 3
0	64	Soprano Sax	0	81	Saw Lead
0	65	Alto Sax	1		Saw Wave
0	66	Tenor Sax	8		Doctor Solo
0	67	Baritone Sax	11		Sawtooth Stab
0	68	Oboe	12		Saw Lead 2
0	69	English Horn	13		Saw Lead 3
0	70	Bassoon	0	82	Synth Calliope
0	71	Clarinet	0	83	Chiffer Lead
0	72	Piccolo	0	84	Charang
0	73	Flute	0	85	Solo Vox
0	74	Recorder	0	86	5th Saw Wave
0	75	Pan Flute	0	87	Bass & Lead
24		Tin Whistle	0	88	Fantasia
25		Tin Whistle Nm	11		Harpsi Pad
26		Tin Whistle Or	12		Fantasia 2
0	76	Bottle Blow	13		Night Vision
0	77	Shakuhachi			

<i>Bank:</i>	<i>PC:</i>	<i>Preset:</i>	<i>Bank:</i>	<i>PC:</i>	<i>Preset:</i>
0	89	Warm Pad	0	106	Shamisen
11		Solar Wind	0	107	Koto
12		Solar Wind 2	8		Taisho Koto
0	90	Polysynth	0	108	Kalimba
0	91	Space Voice	0	109	Bagpipes
0	92	Bowed Glass	0	110	Fiddle
0	93	Metal Pad	0	111	Shenai
0	94	Halo Pad	0	112	Tinker Bell
0	95	Sweep Pad	0	113	Agogo
0	96	Ice Rain	0	114	Steel Drums
11		Mystery Pad	0	115	Wood Block
0	97	Soundtrack	8		Castanets
0	98	Crystal	0	116	Taiko Drum
1		Synth Mallet	8		Concert Bass Drum
11		Synth Chime	0	117	Melodic Tom
0	99	Atmosphere	8		Melodic Tom 2
0	100	Brightness	0	118	Synth Drum
11		Bright Saw Stack	8		808 Tom
0	101	Goblin	0	119	Reverse Cymbal
0	102	Echo Drops	11		Cymbal Crash
2		Echo Pan	12		Tambourine
0	103	Star Theme	0	120	Fret Noise
0	104	Sitar	1		Cut Noise
0	105	Banjo	2		String Slap

<i>Bank:</i>	<i>PC:</i>	<i>Preset:</i>	<i>Bank:</i>	<i>PC:</i>	<i>Preset:</i>
0	121	Breath Noise	0	125	Helicopter
1		Fl. Key Click	1		Car-Engine
11		Filter Snap	2		Car-Stop
0	122	Seashore	3		Car-Pass
1		Rain	4		Car-Crash
2		Thunder	5		Siren
3		Wind	6		Train
4		Stream	7		Jet Plane
5		Bubbles	8		Starship
11		Howling Winds	9		Burst Noise
12		White Noise Wave	0	126	Applause
0	123	Birds	1		Laughing
1		Dog	2		Scream
2		Horse Gallop	3		Punch
3		Bird 2	4		Heart Beat
0	124	Telephone 1	5		Footsteps
1		Telephone 2	0	127	Gun Shot
2		Door Creaking	1		Machine Gun
3		Door	2		Lasergun
4		Scratch	3		Explosion
5		Windchime	11		Interference
			12		Shooting Star

4.2. Percussion Presets

<i>Bank 120 or Ch.10:</i>		
0: Standard 1 Kit	16: Power Kit	32: Jazz Kit
1: Standard 2 Kit	24: Electronic Kit	40: Brush Kit
2: Standard 3 Kit	25: 808/909 Kit	48: Orchestral Kit
8: Room Kit	26: Dance Kit	56: SFX Kit

See the included `Percussion Map.pdf` for a list of the sounds included in each percussion kit.

5. Support Files

Inside the `support` folder that comes with GeneralUser GS, you will find the following files:

- `GeneralUser GS.ins` – A list of all presets, percussion sounds, and MIDI controllers in Cakewalk instrument format, which is supported by most MIDI composing software.
- `MIDI timing test.mid` – This MIDI file can be used to test the accuracy of your MIDI player’s timing. It features a passage from “The HYBRID Collage” (included in the `demo midis` folder) played very quietly while a castanet plays 32nd-notes split across two MIDI channels. Rhythmic subdivisions are played at different velocities:

Note value:	Note-on Velocity
Quarter	127
8th	108
16th	82
32nd	46

The rhythm of the castanets should sound very even.

- `MIDI timing test.ogg` – A recording of the above MIDI file rendered using FluidSynth.

- `Reset GS & SF2.mid` – If you play MIDI files back to back (without restarting the player or using the “panic” feature), parameters set by one MIDI file can sometimes persist into the next, causing strange playback. Playing this MIDI file will reset everything so the next MIDI file you play should sound correct.

There is also a `FluidSynth` folder containing files used by the various tutorials in the sections above.

6. Compressing GeneralUser GS for Web and Mobile Apps

If you wish to further reduce the file size of the GeneralUser GS SoundFont, you can convert it to an SF3 file using [sf3convert](#). This will encode the samples using OGG lossy compression. To avoid audio artifacts, especially on small loops, I recommend setting OGG quality to 0.8 and attenuating the samples by 1 dB using the following command:

```
sf3convert -z -q 0.8 -a -1 GeneralUser-GS.sf2 GeneralUser-GS.sf3
```

This reduces the SoundFont to less than a third of its former size while sounding nearly indistinguishable from the original SF2.

7. Website and Contact

- You can find updates to GeneralUser GS and more of my virtual instruments at <https://www.schriancollins.com>, where you can also contact me if you have any questions.
- If you'd like to be notified of new releases and updates to my virtual instruments, [follow me on X](#).
- For those upgrading from a previous version of GeneralUser GS, [I have created a video](#) showing some of the changes in GeneralUser GS version 2.0.
- If you have found a bug or issue, you can report it on the [GeneralUser GS GitHub page](#).
- If you would like to support my work, please consider [buying me a coffee](#). 😊

Happy music-making!

--*Chris*