Sound Blaster AWE 32/64 HOWTO

Table of Contents

Sound Blaster AWE 32/64 HOWTO 1
by Marcus Brinkmann < Marcus.Brinkmann@ruhr−uni−bochum.de> 1
1. Introduction 1
2. Before you start 1
3. How to install SB AWE sound support 1
4. Testing the Sound Driver 1
5. AWE Driver Software 2
6. Appendix 2
1. Introduction 2
1.1 Acknowledgments 2
1.2 Revision History 2
1.3 New versions of this document 3
1.4 Feedback 3
1.5 Distribution Policy 3
2. Before you start 4
2.1 Introduction 4
2.2 Some general notes about the SB AWE cards 4
2.3 Some general notes about the Plug and Play cards 4
2.4 Some general notes about loadable kernel modules 5
2.5 Some general notes about the kernel sound driver 5
3. How to install SB AWE sound support 6
3.1 Things you will need 6
3.2 Getting started 6
3.3 Compiling the kernel 7
3.4 Reboot 9
4. Testing the Sound Driver 9
4.1 /proc/devices, /dev/sndstat 9
4.2 Output − The Raw Audio Device 11
4.3 Output − The OPL−2/OPL−3 Synthesis 11
4.4 Output − The Wave Table Synthesis 11
4.5 Mixing 12
4.6 Input − Sampling with the Raw Audio Device 12
4.7 The MIDI Port 12
5. AWE Driver Software 12
5.1 sfxload 13
5.2 drvmidi 13
6. Appendix 13
6.1 Additional Information 13
6.2 Sources 14
6.3 Sample isapnp.conf 15

Sound Blaster AWE 32/64 HOWTO

by Marcus Brinkmann < `Marcus.Brinkmann@ruhr-uni-bochum.de`>

v1.2, 11 January 1998

This document describes how to install and configure a Sound Blaster 32 (SB AWE 32, SB AWE 64) card from Creative Labs in a Linux System using the AWE Sound Driver Extension written by Takashi Iwai. It also covers some special tools and players for the SB AWE series. Reference system is a Debian GNU/Linux System, but every other Linux Distribution should also work.

---

1. Introduction

- 1.1 Acknowledgments
- 1.2 Revision History
- 1.3 New versions of this document
- 1.4 Feedback
- 1.5 Distribution Policy

2. Before you start

- 2.1 Introduction
- 2.2 Some general notes about the SB AWE cards
- 2.3 Some general notes about the Plug and Play cards
- 2.4 Some general notes about loadable kernel modules
- 2.5 Some general notes about the kernel sound driver

3. How to install SB AWE sound support

- 3.1 Things you will need
- 3.2 Getting started
- 3.3 Compiling the kernel
- 3.4 Reboot

4. Testing the Sound Driver

- 4.1 `/proc/devices, /dev/sndstat`
- 4.2 Output – The Raw Audio Device
- 4.3 Output – The OPL–2/OPL–3 Synthesis
- 4.4 Output – The Wave Table Synthesis
- 4.5 Mixing
- 4.6 Input – Sampling with the Raw Audio Device
- 4.7 The MIDI Port

5. AWE Driver Software

- 5.1 `sfxload`
- 5.2 `drvmidi`

6. Appendix

- 6.1 Additional Information
- 6.2 Sources
- 6.3 Sample `isapnp.conf`

---

1. Introduction

This is the Sound Blaster AWE HOWTO. It gives you detailed information about getting the most out of your Sound Blaster 32 or better, including Wave Table synthesis. This document covers all SB cards up from the SB 32.

The Sound Blaster series is well−known in the DOS and Windows community, and a lot of Linux users want to use this sound card under Linux as well. Unfortunately, Creative Labs only provides a sound driver for Windows and DOS, so it is not trivial to install and use a SB card under Linux. This document tries to describe how one can use the features of the SB AWE series under a Linux environment.

1.1 Acknowledgments

This documents contains information I got from the AWE Driver FAQ and the ISA PnP FAQ. See section Sources for author and location of this documents. Thanks to the authors, the SB AWE support is possible.

A lot of essential work was done by Hannu Savolainen, who developed the sound driver that comes with the Linux kernel. Thank you!

I want to thank Nicola Bernardelli for testing the AWE64 stuff. Without him, many errors would have remained undetected.

Thanks to the SGML Tools package, this HOWTO is available in several formats, all generated from a common source file.

1.2 Revision History

Version 1.0

first version

Version 1.1

corrected spelling (thanks Curt!), added version requirement for `isapnp`, now available via sunsite and lots of mirrors

Version 1.2

French version now available!, minor corrections, lots of spelling fixed (thanks to ispell)

1.3 New versions of this document

The latest version can be found on my Home Page, namely at Sound Blaster AWE HOWTO.

New versions of this document will be uploaded to various anonymous ftp sites that archive such information including ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/mini.

Hypertext versions of this and other Linux HOWTOs are available on many World−Wide−Web sites, including http://sunsite.unc.edu/LDP/. Most Linux CD−ROM distributions include the HOWTOs, often under the `/usr/doc` directory.

Thanks to Arnaud Launay zoro@lsol.tm.fr, a French version of this document is now available! The latest version can be found via http or anonymous ftp.

If you make a translation of this document into another language, let me know and I'll include a reference to it here.

1.4 Feedback

I rely on you, the reader, to make this HOWTO useful. If you have any suggestions, corrections, or comments, please send them to me, Marcus.Brinkmann@ruhr−uni−bochum.de, and I will try to incorporate them in the next revision.

i would especially welcome information about the startup procedure (boot scripts etc.) of other famous Linux distributions, e. g. Red Hat or SuSE Linux.

If you publish this document on a CD−ROM or in hardcopy form, a complimentary copy would be appreciated. Mail me for my postal address. Also consider making a donation to the Linux Documentation Project to help support free documentation for Linux. Contact the Linux HOWTO coordinator, Tim Bynum linux−howto@sunsite.unc.edu, for more information.

1.5 Distribution Policy

2. Before you start

2.1 Introduction

This document tries to help you install and use a Sound Blaster AWE 32 or Sound Blaster AWE 64 from Creative Labs in your Linux system. The reference system is a Debian GNU/Linux system on an Intel i586 platform, but it should work with any other Linux distribution as well as on every platform that is supported by the Linux sound driver (differences are mentioned where they appear, if any).

Be sure to read the Linux Sound HOWTO (see section Additional Information) carefully. I consider my document a supplement to the Sound HOWTO, and often you can find more information about things I have left out there.

2.2 Some general notes about the SB AWE cards

The SB AWE 32 sound card provides a raw audio device, standard OPL−2/OPL−3 synthesis, a MPU−401 MIDI port and 32 voices EMU 8000 Wave Table synthesis (for an explanation on these and other terms see the Linux Sound HOWTO). One goal of this document is to help you get all these features to work properly.

The SB AWE 64 has the capabilities of the SB AWE 32 and an additional Wave Guide synthesis Creative Labs is especially proud of. The problem for Linux users is, that the additional 32 voices are software generated and output via the raw wave devices. Because Creative Labs sees no market in Linux drivers, a Wave Guide synthesis sound driver is only available for Windows 3.1 and Windows 95.

This means that, from a Linux user's point of view, the SB AWE 32 and SB AWE 64 are almost identical. From now on I will only refer to the SB AWE in general and will only mention differences where they appear (if any).

2.3 Some general notes about the Plug and Play cards

Most modern cards for the Intel platform are ISA PnP cards, which is an abbreviation for ``Plug and Play''. This means, that the card has to be configured by the operation system, and this has to be done through an initialization routine at boot time. In general, there are at least three possible ways to do this:

1. You have a PnP Bios, which means that your Bios is aware of PnP cards and can configure them. If you think you can use all the features of your SB AWE PnP just because you have a PnP Bios, you are out of luck. Even if the Bios claims to support PnP cards it only initializes a subset of the ports and addresses used by your sound card. You will probably be able to play raw wave data, but you won't be able to play MIDI music, for example. For this reason, a PnP Bios is not an option.
2. You have an operating system that supports PnP cards. The current stable Linux kernel (2.0.x) is not such an operating system, so we have to wait for future kernel releases, that will support PnP devices.
3. You have a special program, started at boot time, that initializes your PnP cards. This is the way we do it.

The most commonly used software to initialize PnP cards under Linux are the ``isapnptools'' (see section Sources and section Getting Started). They provide a predictable way to configure all ISA PnP cards in your machine, not only your sound card.

2.4 Some general notes about loadable kernel modules

Some device drivers can be built as modules instead of compiling them into the kernel. You can find more information about modules in the Kernel HOWTO and the Module HOWTO (see section Additional Information).

If you have a PnP card, you *must* install sound support as a loadable kernel module. This means, that you can't build the sound driver into the kernel, but you will have to build it as a module that can be loaded into the kernel at runtime. This is because the kernel will be installed before your ISA PnP card can be configured, and your sound driver has to be loaded after your ISA PnP card is configured.

The sound kernel module can be loaded manually via `insmod sound` or `modprobe -a sound` or in the appropriate boot script of your Linux system (in Debian, it is sufficient to append a single line containing `sound` to `/etc/modules`). Another approach is to launch `kerneld`, a daemon that installs and removes kernel modules as needed.

Note that `kerneld` may not be the best solution for the AWE sound driver module, because it takes time to load the module in the kernel, especially if you want to use Wave Table synthesis and load big Sound Font banks, which you have to do each time after inserting the module. Because `kerneld` removes unused modules after one minute by default, it is perhaps better to insert the sound module manually or at boot time. Note that inserting the sound module manually or at boot time does prevent `kerneld` from removing it when it is idle. By the way: You can manually insert the sound module and use `kerneld` at the same time. The two methods don't conflict, but `kerneld` does not care about the sound module anymore.

This is especially useful if your mixer settings get disturbed after removing and reinstalling the module. A solution for this problem will be given in section Mixing (there is described how you can start your mixer automatically when the sound module gets inserted). However, it takes time for `kerneld` to load the module, to load the sound font bank and to start your mixer, and for this and other reasons it is better to install the sound module at boot time and not let `kerneld` remove it.

2.5 Some general notes about the kernel sound driver

You can install sound support in the kernel as a built−in or as a loadable module. If you have a PnP card, you have to install sound support as a module, because the PnP card needs to be initialized via the isapnptools before the module gets installed.

As you can imagine, you will have to recompile the kernel. I will give you a few hints about it below. For now, let's talk about the sound support in the kernel source. The kernel ships with the Free (Lite) Version of the OSS (USS) sound driver. The current version of this driver (3.5.4) does not support the SB AWE in full, but the SB 16 part of it. So you can have a raw audio device and OPL−2/OPL−3 synthesis if you use it, but you will not be able to play midi music with Wave Table synthesis.

If you want to use the Wave Table device, you can either buy the commercial sound driver from 4Front Technologies (someone please confirm me that it can do it, please), or patch your kernel with the AWE 32 Sound Driver Extension by Takashi Iwai. The former is beyond the scope of this document, I assume you want to use the latter.

The AWE32 Sound Driver Extension (see section Sources) is published under the GNU copyright license and ships with a number of tools to make use of the EMU 8000 wave synthesis of the SB AWE cards.

The `awedrv` software is already included in newer kernel source trees (starting from some 2.1.x kernels, but perhaps you want to upgrade the sources in your kernel tree, they may be old.

3. How to install SB AWE sound support

3.1 Things you will need

These are the requirements for SB AWE support under Linux:

- a functional Linux system (e.g. the Debian GNU/Linux distribution),
- a SB AWE 32 or compatible card (e.g. SB32, SB32 PnP, SB AWE64, ...),
- the sources of the Linux kernel, including the sound driver OSS/Free (normally included, check in `/usr/src/Linux/drivers/sound/Readme`),
- the AWE 32 sound driver extensions.

If you have a PnP card, you will also need:

- the isapnptools software package.

Look in section Sources for information where you can obtain these programs.

If you have a Debian GNU/Linux system, this means that you will need the packages `kernel-source-<version>`, `awe-drv` and perhaps the package `isapnptools`. You will want some of the other `awe-*` packages although they are not essential. I strongly recommend the `kernel-package` for easy kernel compiling and installing.

3.2 Getting started

Let's assume you have properly installed your card in a socket on your motherboard and perhaps already tested it under a DOS or Windows environment.

The next step is to initialize the card at boot time if and only if it is a PnP card. Follow the documentation in the source package to compile and install `isapnptools` (or just use the Debian binary package), and make a

```
# pnpdump > /etc/isapnp.conf
```

as root. This will generate a hopefully proper configuration file for your PnP cards, but with all devices commented out. Please edit this file carefully, and compare the values for DMA channels, IO bases and interrupts with the configuration of the cards under an DOS or Window environment if possible (with Win95, look at the resources used by the card under `resources` in the `device manager`). If you have problems, read the documentation that ships with the `isapnptools` carefully.

CAUTION: `isapnptools` often fails to detect all three I/O ports of the SB AWE Wave Table device. Please check carefully the `WaveTable` entry in your `isapnp.conf` against the example at the end of this

HOWTO. Adjust the I/O base addresses if necessary.

CAUTION: According to the `isapnp-faq`, sometimes `isapnp` fails to program the number of the logical device. If you encounter error messages like the following:

```
Error occurred executing request 'LD 2' on or around line...
```

then try adding `(VERIFYLD N)` at the top of `isapnp.conf`. You have to use at least version 1.10 for this to work. If you can't use version 1.10 or newer, you can also POKE the logical device numbers directly. Please refer to the `isapnp-faq` for more information on this approach. If it doesn't work for you, please contact me (and the `isapnp` people will be interested, too, I would think).

CAUTION: Make sure that the last line is `(WAITFORKEY)`, that will sometimes be omitted by older versions of `pnpdump`.

A sample `isapnp.conf` for only one PnP card (the sound card) can be found at the end of the document (see section Sample <tt>isapnp.conf</tt>).

If you have a Debian system, no further adjustments are necessary. `isapnp` will be started at boot time in `/etc/init.d/boot` with this snippet of script code, that you may want to include in your boot scripts:

```
# Configure the isa plug and play boards before loading
# modules. Need to do this before loading modules to get
# a chance of configuring and starting PnP boards before
# the drivers mess all this up.
#
if [ -x /etc/init.d/isapnp ]
then
  /etc/init.d/isapnp start
fi
```

where `/etc/init.d/isapnp` is

```
#! /bin/sh
# /etc/init.d/isapnp: configure Plug and Play boards
test -x /sbin/isapnp || exit 0
/sbin/isapnp /etc/isapnp.conf
exit 0
```

If you have another Linux distribution, you are on your own. I don't know what to do (anybody willing to submit more specific information?). Just make sure that isapnp gets started *before* the modules will be loaded (see below).

3.3 Compiling the kernel

Before recompiling the kernel, you have to apply the AWE Driver Extension to the sound driver. Even if your kernel source tree already includes the `awedrv` extension (check `/drivers/sound/lowlevel/` for that), you may want to upgrade the `awedrv` software. Follow the installation instructions in the `awedrv` source directory. In brief, you have to run an installation script that applies the patches to the kernel sources.

Be careful if you have upgraded the kernel source tree after running the install script. The script just checks if a certain file exists − if it exists, it does not apply the necessary patch. You are well advised to remove the file `drivers/sound/lowlevel/awe_wav.c` before running the script after upgrading the kernel source.

Next you have to configure the kernel for sound support. I hope you know a bit about kernel compiling; see the Sound HOWTO and the Kernel HOWTO for details. Go in the source directory of your kernel sources (`/usr/src/linux` for example), and do

```
# make config
```

or `make menuconfig` or `make xconfig`. Then you have to configure your kernel in the standard way. Use this opportunity to make a small and powerful kernel, especially designed for your system. Be sure to answer `Enable loadable module support` with `Y`, if you want to install the sound driver as a loadable module (a *must* if you have a PnP card), but I'm sure you want to do it anyway.

At one point, you will be asked if you want sound card support. You are free to answer with `Y` or with `M` if you have *not* a PnP card. You *must* answer with `M`, for module, if you have a PnP card. You have to compile sound card support as a module if you have a PnP card because the PnP card has to be initialized before the module gets loaded.

The following questions you should answer with `Y`, all other with `N`:

```
Sound Blaster (SB, SBPro, SB16, clones) support (CONFIG_SB) [Y/n/?]
Generic OPL2/OPL3 FM synthesizer support (CONFIG_ADLIB) [Y/n/?]
/dev/dsp and /dev/audio support (CONFIG_AUDIO) [Y/n/?]
MIDI interface support (CONFIG_MIDI) [Y/n/?]
FM synthesizer (YM3812/OPL−3) support (CONFIG_YM3812) [Y/n/?]
lowlevel sound driver support [Y/n/?]
AWE32 support (CONFIG_AWE32_SYNTH) [Y/n/?]
```

Only the latter is actually for the Wave Table synthesis. The others are SB 16 options from the OSS/Free sound driver.

In addition, you have to configure the sound card I/O port. Look at the `isapnp.conf` file for hints, if you have one. For me, the following default values are sufficient. Note that the default value from the kernel configuration script may be wrong (especially the values `SBC_IRQ` and `SB_MPU_BASE` seem to be incorrect for most cards).

```
I/O base for SB Check from manual of the card (SBC_BASE) [220]
Sound Blaster IRQ Check from manual of the card (SBC_IRQ) [5]
Sound Blaster DMA 0, 1 or 3 (SBC_DMA) [1]
Sound Blaster 16 bit DMA (_REQUIRED_for SB16, Jazz16, SMW) 5, 6 or 7
(use 1 for 8 bit cards) (SB_DMA2) [5]
MPU401 I/O base of SB16, Jazz16 and ES1688 Check from manual of the card
(SB_MPU_BASE) [330]
SB MPU401 IRQ (Jazz16, SM Wave and ES1688) Use −1 with SB16 (SB_MPU_IRQ) [-1]
```

Now recompile the kernel. Debian users should use the `kernel-package`. This package makes the kernel compile as easy as installing a debian package. Look at the documentation in `/usr/doc/kernel-package/`. Here is a hint:

```
# make-kpkg clean
# make-kpkg -revision custom.1.0 kernel_image
```

and then `dpkg -i /usr/src/kernel-image-2.0.29_custom.1.0_i386.deb`.

If you have another Linux distribution, follow the standard way for compiling a new kernel. Don't forget `make modules` and `make modules_install`. Look at the Sound HOWTO and perhaps the Kernel HOWTO for more information.

3.4 Reboot

After installing the new kernel, you should do a reboot (be sure to have a functional boot disk at hand). Cross you fingers.

If you have a PnP card be sure to launch isapnp either in a boot script (as described above) or manually:

```
# /sbin/isapnp /etc/isapnp.conf
Board 1 has Identity 74 00 00 e3 10 48 00 8c 0e:  CTL0048 Serial No 58128
[checksum 74]
```

Now you can install the sound driver, if you have compiled it as a module:

```
# modprobe -a sound
AWE32 Sound Driver v0.3.3e (DRAM 2048k)
```

If you think the memory detection was not correct (I have a report of one who has a AWE64 with 4096k, and ``detected'' have been 28672k), try either to upgrade the `awedrv` software or to specify the amount of memory in the file `/usr/src/linux/drivers/sound/lowlevel/awe_config.h`, for example:

```
#define AWE_DEFAULT_MEM_SIZE  4096   /* kbytes */
```

Sorry, you have to recompile the kernel then (perhaps compiling the modules will be sufficient, but I don't know for sure).

If it works, you may want to have the sound module loaded automatically. You can use `kerneld` (why this is a bad idea is explained in section 1.4) or append a single line containing `sound` to your `/etc/modules` (in Debian) or add `/sbin/modprobe -a sound` to your start–up script.

---

4. Testing the Sound Driver

4.1 `/proc/devices, /dev/sndstat`

If you have built–in sound support, you will get some useful information at boot time. If you have sound support installed as a loadable module, you can get the same information (perhaps after removing the sound module with `modprobe -r sound` first) with

```
# modprobe -a sound trace_init=1
Sound initialization started

<Sound Blaster 16 (4.13)> at 0x220 irq 5 dma 1,5
<Sound Blaster 16> at 0x330 irq 5 dma 0
<Yamaha OPL3 FM> at 0x388
Sound initialization complete
AWE32 Sound Driver v0.3.3e (DRAM 2048k)
```

If you have a `/proc` virtual file system, you can look for the sound device with

```
# cat /proc/devices
Character devices:
[...]
14 sound
[...]
```

Next make sure you have the correct devices installed under `/dev/`. Please look at the Sound HOWTO for details. Then ask `/dev/sndstat` about the status of the sound module:

```
# cat /dev/sndstat
Sound Driver:3.5.4-960630 (Sat Oct 11 19:35:14 CEST 1997 root,
Linux flora 2.0.29 #1 Sat Oct 11 19:12:56 CEST 1997 i586 unknown)
Kernel: Linux flora 2.0.29 #1 Sat Oct 11 19:36:23 CEST 1997 i586
Config options: 0

Installed drivers:
Type 1: OPL-2/OPL-3 FM
Type 2: Sound Blaster
Type 7: SB MPU-401

Card config:
Sound Blaster at 0x220 irq 5 drq 1,5
SB MPU-401 at 0x330 irq 5 drq 0
OPL-2/OPL-3 FM at 0x388 drq 0

Audio devices:
0: Sound Blaster 16 (4.13)

Synth devices:
0: Yamaha OPL-3
1: AWE32 Driver v0.3.3e (DRAM 2048k)

Midi devices:
0: Sound Blaster 16

Timers:
0: System clock

Mixers:
0: Sound Blaster
1: AWE32 Equalizer
```

If you don't have an output like this, perhaps there is an error in your configuration. Go on and see what doesn't work, then go back to step Getting Started, checking everything.

4.2 Output − The Raw Audio Device

Try to get an `.au` file (Sun workstation) or a raw sample file, and do

```
# cat bell.au > /dev/audio
```

resp.

```
# cat sample > /dev/dsp
```

You should hear the content of the file via the `Audio Device 0: Sound Blaster 16 (4.3).`

4.3 Output − The OPL−2/OPL−3 Synthesis

If you want to use the OPL−2/OPL−3 FM synthesis to play MIDI−Files with your sound card, try the program `playmidi` (see Appendix B). Start it with

```
# playmidi −f dance.mid
```

or

```
# playmidi −4 dance.mid
```

The former will give you OPL−2, the latter OPL−3 MIDI music. If you are embarrassed about the sound, don't blame playmidi: It is the FM synthesis that sounds bad.

Imagine you would only have the OSS/Free Sound Driver: Than this is the quality of MIDI Music you can achieve (apart from software synthesis). Fortunately, you have a SB AWE, and you can use the Wave Table capabilities with the AWE Sound Driver Extension.

4.4 Output − The Wave Table Synthesis

The AWE Driver Extension comes with special tools (`awesfx`) to make use of the EMU 8000 Wave Table synthesis. First, you have to load a Sound Font Bank on your card − even if you want to use the ROM samples! You can get the files from your Windows installation − look for files ending in `*.sfb` or `*.sf2`.

The ROM samples can be loaded with `SYNTHGM.SBK`, real samples are in `SYNTHGS.SBK` and `SYNTHMT.SBK`, as well as in `SAMPLE.SBK`. You can get other Sound Font Banks via ftp or www, try the EMU Homepage, the web site from Creative Labs, and look out for the Chaos samples, they are really good (be sure to check out the AWE Driver Web Site).

Try to load the standard GM (ROM) set with

```
# sfxload −i synthgm.sbk
```

and then play a midi file that comes with your SB AWE:

```
# drvmidi dance.mid
```

4.5 Mixing

Get your favorite mixer program and start it. Start a raw audio sample and two MIDI files at the same time, and try out the mixer settings. Play a bit, it should be very easy. Here a list of the devices and how they are called:

Yamaha OPL−3

Synth or FM

AWE32 Driver

Synth or FM (does anyone know a mixer were the last two are separated from each other?)

Sound Blaster 16 (4.13)

PCM or DSP

PC Speaker (still alive)

Spkr

Other mixer settings refer to the CD ROM, possibly connected to the sound card, to Master Volume, Bass, Treble and Recording Level of the different input lines. You can specify which lines should be recorded.

4.6 Input − Sampling with the Raw Audio Device

You can record different sources: a connected CD ROM, a microphone connected to `mic`, and any that you can connect to `line in`. Set the mixer in the appropriate position. Play a sound and record into a file, reading from the Raw Audio Device, for example:

```
# cdplay
# dd bs=8k count=5 </dev/dsp >music.au
5+0 records in
5+0 records out
# cat music.au >/dev/dsp
```

records and plays five seconds of audio from the input device.

4.7 The MIDI Port

Sorry, no information about MIDI port yet!

---

5. AWE Driver Software

5.1 `sfxload`

You can load samples in your DRAM on the sound card with the `sfxload` tool. Note that you can only load one sample file per bank, with bank 0 as default. So, after `sfxload synthgs.sbk`, the only samples on your sound card are the GS samples. If you want to load additional Sound Font Banks, you have to use the `-b` option, for example:

```
# sfxload synthgs.sbk
# sfxload -b1 sample.sbk
# drvmidi sfx.mid
```

You can load a default Sound Font Bank automatically with installing the module. Just append a line like

```
post-install sound /usr/bin/sfxload synthgm.sbk
```

to your `/etc/conf.modules` file.

See the documentation for more details about `sfxload`.

5.2 `drvmidi`

With the `drvmidi` program, you can use your AWE Driver to play MIDI files. Just specify the name of your midi file after the command:

```
# drvmidi waltz.mid
```

See the documentation for more details about `drvmidi`

---

6. Appendix

6.1 Additional Information

The Linux Sound HOWTO

Author: Jeff Tranter, < `jeff_tranter@pobox.com`>

Last Version: v1.17, 4 August 1997

Please take a good look at the Sound HOWTO (available from sunsite.unc.edu and other sites). It contains a bulk of data about compiling the kernel with sound support, and explains a lot about the sound devices, applications etc.

I understand this HOWTO as an addition to the Sound HOWTO.

The Linux Sound Playing HOWTO

Author: Yoo C. Chung, < `wacko@laplace.snu.ac.kr`>

Last Version: v1.4, 13 May 1997

You should also look at the Sound Playing HOWTO. It tells you about the various sound formats and applications to play them.

The AWE Driver FAQ

Author: Takashi Iwai < `iwai@dragon.mm.t.u-tokyo.ac.jp`>

Source: http://bahamut.mm.t.u–tokyo.ac.jp/~iwai/awedrv/awedrv–faq.html

If you have problems installing the AWE Driver or using the Wave Table synthesis of your Sound Blaster card, then look here.

The ISA PnP FAQ

Author: Peter Fox < `fox@roestock.demon.co.uk`>

Source: http://www.roestock.demon.co.uk/isapnptools/isapnpfaq.html

If you have problems configuring your ISA PnP card, then this is your book of wisdom.

6.2 Sources

`isapnptools`

Author: Peter Fox < `fox@roestock.demon.co.uk`>

Last Version: 1.11

Source: http://www.roestock.demon.co.uk/isapnptools/index.html

If mailing patches, bug reports or comments, please put 'isapnp' somewhere in the subject line, and mail to isapnp@roestock.demon.co.uk.

`awedrv`

Author: Takashi Iwai < `iwai@dragon.mm.t.u-tokyo.ac.jp`>

Last Version: 0.4.2c

Source: http://bahamut.mm.t.u–tokyo.ac.jp/~iwai/awedrv/

Linux kernel

Author: Linus Torvald and many other

Last Version: At the moment, you should use 2.0.29

Source: everywhere, where you can obtain Linux `:)`

OSS/Free

Author: Hannu Savolainen (Please check http://www.4front−tech.com/usslite before mailing me).

Last Version: I have 3.5.5−beta1

Source: with the Linux kernel source package

Information: http://www.4front−tech.com/usslite or European mirror.

6.3 Sample `isapnp.conf`

In the ISA PnP configuration file created by pnpdump, the single devices of your PnP cards appear as sections. In the output below, one card was detected (the sound card), with four logical devices:

- `LD 0`: Audio Device
- `LD 1`: IDE Device
- `LD 2`: WaveTable Device
- `LD 3`: Gameport Device

I left `LD 1` unconfigured, because I have no CD ROM attached to my sound card IDE port. If you have no IDE port on your SB, then `LD 1` will be the Gameport Device and `LD 3` will not appear.

Please refer to section Getting started for further (important!) information about this file.

```
# $Id: pnpdump.c,v 1.9 1997/06/10 21:37:32 fox Exp $
# This is free software, see the sources for details.
# This software has NO WARRANTY, use at your OWN RISK
#
# For details of this file format, see isapnp.conf(5)
#
# For latest information on isapnp and pnpdump see:
# http://www.roestock.demon.co.uk/isapnptools/
#
# Compiler flags: −DREALTIME −DNEEDSETSCHEDULER

(READPORT 0x0203)
(ISOLATE)
(IDENTIFY *)

# Try the following if you get error messages like
# Error occurred executing request 'LD 2' on or around line...

#(VERIFYLD N)

# ANSI string −−>Creative SB32 PnP<−−

(CONFIGURE CTL0048/58128 (LD 0
#     ANSI string −−>Audio<−−

  (INT 0 (IRQ 5 (MODE +E)))
  (DMA 0 (CHANNEL 1))
  (DMA 1 (CHANNEL 5))
  (IO 0 (BASE 0x0220))
```

```
  (IO 1 (BASE 0x0330))
  (IO 2 (BASE 0x0388))

  (ACT Y)
))

(CONFIGURE CTL0048/58128 (LD 1
#     ANSI string -->IDE<--

# (INT 0 (IRQ 10 (MODE +E)))
# (IO 0 (BASE 0x0168))
# (IO 1 (BASE 0x036e))

# (ACT Y)
))

(CONFIGURE CTL0048/58128 (LD 2
#     ANSI string -->WaveTable<--

  (IO 0 (BASE 0x0620))
  (IO 1 (BASE 0x0A20))
  (IO 2 (BASE 0x0E20))

  (ACT Y)
))

(CONFIGURE CTL0048/58128 (LD 3
#     ANSI string -->Game<--

  (IO 0 (BASE 0x0200))
  (ACT Y)
))

# Returns all cards to the "Wait for Key" state
(WAITFORKEY)
```