

GETTING STARTED

With FMOD Ex Programmer's API for Linux



LEGAL NOTICE

The information in this document is subject to change without notice and does not represent a commitment on the part of Firelight Technologies. This document is provided for informational purposes only and Firelight Technologies makes no warranties, either express or implied, in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results of the use of this document remains with the user. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Firelight Technologies.

© 2010-2012 Firelight Technologies Pty. Ltd. All rights reserved.

Other product and company names mentioned herein may be the trademarks of their respective owners.

CONTENTS

Contents

Introduction	4
Support Resources	5
API documentation	5
Forums	5
Email.....	5
Videos.....	5
Installation	6
Formats not supported	6
ALSA driver arguments.....	6
Output mode selection	6
Troubleshooting.....	7
Pulsating tone is suddenly audible	7

Introduction

Welcome to the FMOD Ex Programmer's API for Linux, the quickest and easiest way to get great sound and music into your Linux games. This document will show you how to get started implementing FMOD Ex in your game by pointing you in the direction of detailed API documentation and support resources. While the FMOD Ex Programmer's API presents the same interface on all platforms, each platform does have its own unique features and limitations - Linux-specific features/limitations will be listed here along with any hints and tips for getting the most out of FMOD Ex on Linux.

Have fun implementing great audio and drop us a line some time,

The FMOD Team
Melbourne, Australia
www.fmod.org

Support Resources

API documentation

Detailed API documentation can be found in the “documentation” directory/folder of your FMOD Ex Programmer’s API installation. This documentation is your main reference for information on FMOD Ex API classes and functions.

Forums

<http://www.fmod.org/forum>

This should be your first port of call for further FMOD information and questions on implementation. If you have a question related to FMOD, chances are someone else has already asked it. The FMOD forums are free for all FMOD users and are monitored by the FMOD team as well as being home to a strong community of FMOD developers, from student first-timers to top-level professionals working on games that are household names.

Email

support@fmod.org

This is our main technical support line. It’s monitored directly by the FMOD team and we aim to answer all emails within 24 hours. It’s free for all FMOD users and your issues will be addressed directly by the guys who wrote the code. If you can’t find an answer to your problem on the FMOD forums, shoot us an email and we’ll get right onto it.

Videos

<http://www.youtube.com/FMODTV>

The FMOD YouTube channel contains a growing number of videos of tutorials relating to FMOD and FMOD Designer. This channel is being added to all the time, so be sure to check back regularly.

Installation

Link the following library into your project:

- Use **api/lib/libfmodex.so** to use the FMOD Ex API.

Formats not supported

WMA is the only common PC file format not supported in FMOD Ex for Linux. This is because FMOD uses a Windows codec to be able to decode WMA. This codec is proprietary and owned by Microsoft, it is not cross platform.

ALSA driver arguments

If a particular device needs arguments, for instance you wish to target device "hw:1:0" (the second sound card, first device). You can do this by selecting the "hw" device via **System::setDriver**, then set the appropriate argument via the **FMOD_LINUX_EXTRADRIVERDATA** structure.

Arguments can be passed in with the **System::init** 'extradriverdata' parameter. This **extradriverdata** parameter is a pointer to a **FMOD_LINUX_EXTRADRIVERDATA** structure found in `fmodlinux.h`. You can use this structure to specify the arguments for both output and record drivers.

Output mode selection

By default if you do not specify the output mode via `System::setOutputType` FMOD will try to determine the best choice for your system. The order of selection is as follows, note that ESD can only be selected manually:

1. **PulseAudio**, detected via "pulseaudio --check".
2. **ALSA**, detected via checking for the existence of "libasound.so.2".
3. **OSS**, fallback if neither PulseAudio nor ALSA is available.

Troubleshooting

Find solutions for common platform specific issues here:

A quick note. Use the logging version of FMOD to get information in the tty or output log file.

Pulsating tone is suddenly audible

This is a fatal warning from FMOD's mixer. It means the mixer tried to allocate some memory and failed. Because of unexpected behavior at this point, the mixer sends a pulsating sine wave out through the speakers to let you know of this fact.

The solution for this is to reduce memory usage or provide more memory to FMOD, then restart the application.

Note that the tty/log output will display out of memory error messages, and `System::setCallback` can be used in the API to catch out of memory errors with `FMOD_SYSTEM_CALLBACKTYPE_MEMORYALLOCATIONFAILED`.