GETTING STARTED

With FMOD Ex Programmer's API for Google Native Client

UPDATED : MAY 2012

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.

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

CONTENTS

Contents

Introduction

Welcome to the FMOD Ex Programmer's API for Google Native Client, the quickest and easiest way to get great sound and music into your Google Native Client 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 - Google Native Client-specific features/limitations will be listed here along with any hints and tips for getting the most out of FMOD Ex on the Google Native Client.

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

Libraries

FMOD comes with static C/C++ libraries for X86 and X64 and both pepper v14 and pepper v15.

- **libfmodex.a** for general development.
- **libfmodexL.a** for the same library, but with debug logging which can help to determine any problems if they exist.

The library location follows the format **api/lib/$(PEPPER_VERSION)/$(TOOLCHAIN)/$(ABI)**

Initialization

When initializing FMOD you must supply a handle to the instance of the Native Client module that will own the audio resource.

Create a FMOD_NACL_EXTRADRIVERDATA structure and set the instance member to the value returned by pp_instance().

```
#include "fmodnacl.h"

class Example : public pp::Instance
{
    virtual bool Init(uint32_t argc, const char *argn[], const char * argv[])
    {
        // ...

        FMOD_NACL_EXTRADRIVERDATA extraDriverData;
        memset(&extraDriverData, 0, sizeof(FMOD_NACL_EXTRADRIVERDATA));
        extraDriverData.instance = pp_instance();
        result = mSystem->init(32, FMOD_INIT_NORMAL, &extraDriverData);

        // ...
    }
}
```

Running Examples

The examples are built using Make. Make sure you have the environment variable **NACL_SDK_ROOT** set to the installation directory for the Google Native Client SDK. Specify the target architecture using the configuration variable ABI="x86-32" or ABI="x86-64". Once the executable has been built copy the program folder plus the **media** and **common_js** folder to the root directory of a web-server and load the HTML file in the browser.

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`