

NativeFmod Project

You want to use Fmod (www.fmod.org) in Java ? I've created NativeFmod for you.
Copyright © 2004 Jérôme JOUVIE (jouvieje)

Created by Jérôme JOUVIE (Jouvieje)
E-mail: jerome.jouvie@gmail.com
<http://topresult.tomato.co.uk/~jerome/>
<http://www.esstin.uhp-nancy.fr/~jouvieje/>

Version: NativeFmod version 3.11
Fmod v3.74 supported (3.74.1 for Mac)

-> EVOLUTION OF THE VERSION 3.11

- * Better exception handle in callbacks.
- * Bugs with particular characters is now fully corrected. I forgot to test some characters when I've tried to correct this bug in 2.22.
Special thanks to Henning Lippke for his help to correct this problem.

Now, I've think that NativeFmod don't need a lot of more changes.
Please contact me you remarks bugs.

-> EVOLUTION OF THE VERSION 3.1

- * By writting NativeFmodEx, I've remarked that I make some mistakes in v3.0 and older ! I've corrected these problems in 3.1.
Here are visible changes :
 - x long, long[] and LongBuffer types has been replaced respectively with int, int[] and IntBuffer.
- * Examples updated. Some bugs corrected in the examples that use callbacks.
- * Change in FSOUND_OPENCALLBACK, becomes :
 - x public Pointer FSOUND_OPENCALLBACK(String name);
- * Bug fixed in final 3.1 (03/05/2005) BufferUtils.toBuffer(...).

Final 3.1 released on 03/05/2005.

-> EVOLUTION OF THE VERSION 3.0

- * I've change all C*** classes. Some of them (C***Array and CString) are replaced by NIO Buffers (look at NIO Api in Sun's documentation). CObject and CPointer are replaced by ObjectPointer and Pointer.
 - C***Array, CString
 - X CObject, CPointer
 - + ObjectPointer and Pointer
- * I've also added some utilities to manipulates Buffer and Pointers. To know more about this, look at the following classes in NativeFmod Javadoc :
 - + BufferUtils and PointerUtils
- * Bug fixed so -Djava.library.path can be used with the Win32 support. Loading of libraries are now done using Fmod.loadLibraries() & FmodDyn.loadLibraries() :
 - + loadLibraries (in Fmod class)
 - + loadLibraries (in FmodDyn class)
 - + InitException
- * Fmod samples ane now all ported.

-> EVOLUTION OF THE VERSION 2.22

- * Added in the classes NativeFmod and FMOD_INSTANCE to allow pass the file data as a parameter (look at stream example) :
 - + FSOUND_STREAM FSOUND_Stream_Open(CPointer, long, int, int)
- * FSOUND_Stream_AddSyncPoint and FSOUND_CD_SetPlayMode signatures are changed (same than in NativeFmod class). The NATIVEFMOD_GetVersion link was broken, you can now use it.
- * CObject class is now finished. I've also "review" all the C... classes (CByteArray ...) to clean and optimize them. I think that it is the last modification into these classes.
- * Documentation added in missing classes. I've tried to explain you in the documentation how works C... classes and how to use them (it was a little confused in previous versions). Tell me what you think about new comments of these classes (I like to hear you !).
- * Fmod sample finished :
stream and simple(remains a little optional thing).

-> EVOLUTION OF THE VERSION 2.21

- * The version 3.74 is now supported by the version 2.21.
- * Changes on callbacks :
In previous versions callback was abstract classes, now there are interfaces. Interfaces are more flexible to manipulate than abstract classes.
Sorry for those that use previous version of NativeFmod, you have to replace the method name (that is to say callback) by the interface name.
 - x FMUSIC_CALLBACK, FSOUND_ALLOCCALLBACK, FSOUND_CLOSECALLBACK...
- * FSOUND_Stream_GetTagField works well. Signature of this method becomes :
 - x boolean FSOUND_Stream_GetTagField(FSOUND_STREAM, int, int[], CString, CPointer, int[])
- * FSOUND_CD_SetPlayMode links was broken. You now can use this method.
- * Changes into all C***Array classes, look at the Java documentation to view these changes.
- * Ameliorations internaly to the library.
- * Mac support (for Fmod 3.74.1) added.
- * Fmod samples and tests are updated to be run under Mac platform. I've added script to execute them under Linux and Mac (sh files).
- * Project becomes an Open Source project.

-> EVOLUTION OF THE VERSION 2.2

- * I've begin to port Fmod samples. Fmod samples now fully ported :
3d, cdda, cddarip, fsb, multiple, netstream, simplest and stream2
- * Multiple output should works well now since the version 17082004. Use the classes NativeFmodDyn and FMOD_INSTANCE.
 - + NativeFmodDyn (class)
 - FMOD_CreateInstance (method in NativeFmod class)
 - FMOD_FreeInstance (method in NativeFmod class)
- * equals method added for all fmod type. Use this method to know if two Fmod objects are the same
 - + equals (method)
- * After a lot of failure tests and many hours lost :-(, I finally achieve :-) what I would to do with the CVoidPointerPointer class.
With the new changes, you can use FSOUND_Stream_FindTagField, FSOUND_Stream_GetTagField and FSOUND_Sample_Lock like in the Fmod samples. Look at cdda fmod sample (in Music.NativeFmod.FmodSamples) it is now fully ported.

- CVoidPointerPointer (class)
- CVoidPointer (class)
- + CPointer (I make this class public)

These classes allows you to access values stroed in a C++ object :

- + CString (class)
- + CByteArray, CCharArray, CShortArray, CIntArray, CLongArray, CFloatArray, CDoubleArray (classes)

- * Windows Dll are smaller : NativeFmod.dll+NativeFmodDyn.dll=228ko instead of 500ko !!!
I think it will runs a little faster (on slow computer). An error like "Assertion Failure" (that result in a crash of the Java Virtual Machine) can occure with the previous version. Now, I think this problem is solved.
- * I add a method that return the current platform being used. I added this method because the output depends on the platform in which the program runs.
 - + PLATFORM (field in the class NativeFmod)
 - + PLATFORMS (interface)
- * To simplify the use of fmod constants, I've created an interface that contains all the fmod constants :
 - + FMOD_CONSTANTS (interface)
- * Add access to the version of NativeFmod :
 - + NATIVEFMOD_VERSION (field in NativeFmod class)
 - + NATIVEFMOD_GetVersion (method in NativeFmod class)
- * FSOUND_SEEKCALLBACK signature has changed. Now it is :


```
public int callback(CPointer handle, int pos, byte mode)
```

 instead of


```
public int callback(CPointer handle, int pos, boolean mode)
```
- * A problem in reading a string is now solved.
- * Javadoc contents now the tags links and remarks.

-> EVOLUTION OF THE VERSION 2.13

- * Modifications in FSOUND_REVERB_PROPERTIES and FSOUND_REVERB_PRESETS
- * CFloatArray remove and replaced by float[]
- * CIntPtr remove and replaced by int[]
- * CUnsignedIntPtr remove and replaced by long[]
- * Allow use multiple soundcards outputs :
 - + FMOD_INSTANCE (class)
 - + FMOD_CreateInstance (method in NativeFmod class)
 - + FMOD_FreeInstance (method in NativeFmod class)
- * Get error as string :
 - + FMOD_ErrorString (method in NativeFmod class)
- * Exception managing (mainly NullPointerException) added internaly of the NativeFmod library. I add this to prevent crashes outside the Java VM (Virtual Machine) :
when an exception is occured (mainly NullPointerException) internaly to the NativeFmod library, the exception is treat as a Java exception
- * Warning : package name is changed to : Music.NativeFmod (NativeFmod libraries before the build 20072004 are not compatible with the following versions)

-> EVOLUTION OF THE VERSION 2.12

- * Fmod 3.73 supported :
 - FSOUND_CD_Eject is deprecated, use instead FSOUND_CD_OpenTray
 - + FSOUND_CD_OpenTray (method in NativeFmod class)
- New structure FSOUND_TOC_TAG that defines a CD Table contents.
 - + FSOUND_TOC_TAG (class)
- * NativeFmod javadoc available
- * Separe Fmod constants in different classes
- * Add FSOUND_REVERB_PRESETS and add constructors in the class FSOUND_REVERB_PROPERTIES

with a Fmod preset (FSOUND_REVERB_PRESETS)

* Fmod Callback supported

-> EVOLUTION OF THE VERSION 2.11

* CFloatPointer renamed to CFloatArray

I add constructors, getter and setter in the CFloatArray class

-> End of April 2004

I found the Swig software. Swig is a software that simplify the linking between c/c++ and many language like java. Thanks to this software, I create the first version 2.0 of NativeFmod (not released)

The first public release (version 2.1) is now available online.

-> November 2003

I find the Fmod API in the web and I begin to search how I can use it in my java projects. I begin to creates a little app that call few Fmod methods (about 30) by using JNI (Java Native Interface)

It is the version 1.0 (not released)