Inside MADLibrary 4.01

by Antoine ROSSET 95

16 BD Tranchées

1206 GENEVA

Switzerland

Available for Think C - Symantec - CodeWarrior - MPW - XCMD (HyperCard, MacroMind)

MADLibrary is a collection of routines that you can use to implement music in your applications. You can use MADLibrary to...

- •   play music

- •   play sounds during music

- •   read different formats available in plugs : MAD, MOD, XM, S3M, MTM, etc.

- •   create your own soundtracker editor or player

MADLibrary Installation

To use the MADLibrary functions, install the library in your project and make an #include"RDriver.h" in your file ".c". You'll need 3 files: MAD.h, RDriver.h and MADLibrary.

WARNING

For previous users:

- Version 4.0 doesn't need anymore BGGB resource. Delete it!

- MADLibrary internal music format is MADH, not MADF anymore. You'll have to resave your musics with last version of PlayerPRO to use them with this library.

- MADGetBestDriver replaces GetBestDriver in previous examples.

- Structure names have changed...

MADLibrary Reference

This section serves as a reference to the routines MADLibrary provides.

MADInitLibrary

This function initializes the MADLibrary package.

```
OSErr MADInitLibrary(Str255 PlugsFolderName);
```

DESCRIPTION

The `MADInitLibrary` function is used to initialize the MADLibrary package. `MADInitLibrary` performs some checks to see is MADLibrary can run, and then sets up some internal data structures (it allocates about 20kb). You must call `MADInitLibrary` before calling any other MADLibrary routine. It will also check if there are some

Import/Export plugs available: it checks application directory and the `PlugsFolderName` directory if it exists: you can give a "\p" filename, but not a 0L !

`MADInitLibrary` returns an error code is initialization fails, other wise it returns noErr.

MADDisposeLibrary

This function shuts down the MADLibrary package.

```
void MADDisposeLibrary(void);
```

DESCRIPTION

The `MADDisposeLibrary` function is used to shut down the MADLibrary package. It should be called when the application is finished using MADLibrary to balance the original call to `MADInitLibrary`. At this point no further calls should be made to any MADLibrary routines until `MADInitLibrary` is called again.

MADGetBestDriver

This will check current Mac hardware and fill the MADDriverSettings structure with the best settings for current Mac. This function doesn't call any other functions of MADLibrary.

```
OSErr MADGetBestDriver( MADDriverSettings *driverParam);
```

driverParam   A pointer to your driver settings:

numChn (2-32).   Active tracks, automatically updated when a new music is loaded

outPutBits   8 or 16 bits

outPutRate   Fixed number, by example: rate44Khz, rate22050khz, rate11khz, etc...

outPutMode   MonoOutPut, StereoOutPut or DeluxeStereoOutPut ?

driverMode   This should always be SoundManagerDriver

antiAliasing   Use anti-aliasing filter?

repeatMusic   When music will be over, repeat it?

sysMemory   Allocate memory in application heap(false) or in system heap(true).

Interpolation   Sound Interpolation active?

MicroDelay   Micro delay active? Used only in DeluxeStereoOutPut outPutMode.

MicroDelaySize Micro delay duration (in ms, max 1 sec = 1000 ms)

surround   Surround effect active?

DESCRIPTION

This will check current Mac hardware and fill the MADDriverSettings structure with the

best settings for current Mac. This function doesn't call any other functions of MADLibrary. The common usage is to use it just before MADCreateDriver function.

MADCreateDriver

Use This function will create a new music driver, allowing you to specify the settings to be used.

```
OSErr MADCreateDriver(MADDriverSettings *driverParam);
```

driverParam	A pointer to your driver settings:
numChn (2-32).	Active tracks, automatically updated when a new music is loaded
outPutBits	8 or 16 bits
outPutRate	Fixed number, by example: rate44Khz, rate22050khz, rate11khz, etc...
outPutMode	MonoOutPut, StereoOutPut or DeluxeStereoOutPut ?
driverMode	This should always be SoundManagerDriver
antiAliasing	Use anti-aliasing filter?
repeatMusic	When music will be over, repeat it?
sysMemory	Allocate memory in application heap(false) or in system heap(true).

DESCRIPTION

You have to call this function before calling loading and playing functions. The `MADCreateDriver` function is used to create a new music driver. It is strongly advised to launch this routine at the beginning of your program. See example.c to see parameters how you can perform an automatic set up of driverParam by using `MADGetBestDriver`. This functions allocates about 10kb. You have to call `MADDisposeDriver` if you want to call this function many times.

MADDisposeDriver

This function delete current music driver created with `MADCreateDriver`.

```
OSErr MADDisposeDriver(void);
```

DESCRIPTION

This function delete current music driver created with `MADCreateDriver`. You cannot use loading and playing function after this call (you have to call `MADCreateDriver` again.)

MADLoadMusicFile

This will load a music file into memory. The music file has to be a 'MADH' file, created with PlayerPRO last version.

```
OSErr MADLoadMusicFile(Str255 name);
```

name   Music file name in current directory.

DESCRIPTION

The `MADLoadMusicFile` function is used to load a PlayerPRO music file in memory. You have to set directory to the music file directory, otherwise it will return a file error.

This call is equivalent to `MADImportMusicFile('MADH', filename);`

MADLoadMusicRsrc

This will load a music resource into memory. The music resource has to be a 'MADH' file, created with PlayerPRO last version.

```
OSErr MADLoadMusicRsrc(OSType resType, short resID);
```

resType   Resource type
resID   Resource ID

DESCRIPTION

The `MADLoadMusicRsrc` function is used to load a PlayerPRO music resource in memory. You have to open your resource file before if resource is not in application resources, otherwise it will return a file error.

MADLoadMusicPtr

This will load a music pointer into memory. The music pointer has to be a 'MADH' file, created with PlayerPRO last version.

```
OSErr MADLoadMusicPtr(Ptr musicPtr);
```

musicPtr   A pointer on music data

DESCRIPTION

The `MADLoadMusicPtr` function is used to load a PlayerPRO music pointer in memory. You can dispose your pointer after this call, MADLibrary will not access data on your pointer.

MADImportMusicFile

This will load a music file into memory. This function will check if there is a 'Plug-ins' to load this music.

```
OSErr MADImportMusicFile(OSType type, Str255 name);
```

`OSType`   File type : 'MADH', 'MADF', 'XM' ,'S3M', etc.

DESCRIPTION

The `MADImportMusicFile` function is used to load a PlayerPRO music filte into memory. It will check if there are some Import/Export plugs available for this type of music: it checks application directory and the 'Plugs' directory if it exists.

MADDisposeMusic

This will dipose current music in memory after a load function.

```
OSErr MADDisposeMusic( void);
```

DESCRIPTION

The `MADDisposeMusic` function is used to dispose the current music in memory.

MADPlay

This will play current music in memory.

```
OSErr MADPlay( void);
```

DESCRIPTION

The `MADPlay` function is used to play the current music in memory.

MADStop

This will stop playing current music in memory.

```
OSErr MADStop( void);
```

DESCRIPTION

The `MADStop` function is used to stop playing the current music in memory.

MADReset

This will reset reading position of current music in memory to startup position.

```
OSErr MADReset( void);
```

DESCRIPTION

The `MADReset` function is used to reset reading position of current music in memory to startup position.

MADPlaySndHandle

This will play a sound handle of 'snd ' type on a MADLibrary driver channel.

```
OSErr MADPlaySound( Handle sndRsrc, long channel, long note);
```

`sndRsrc`	Handle on a 'snd ' handle (see Inside Macintosh)
`channel`	Channel ID which will be used to play this sound
`note`	note ID: from 0 (C0) to 95 (B7) or 0xFF (play this sound at normal rate)

DESCRIPTION

The `MADPlaySndHandle` function is used to play a sound 'snd ' on a MADLibrary driver channel. WARNING: This function will change the sndRsrc handle, you will not be able to use it with normal SoundManager functions: you will have to reload it. (It will inverse amplitude of data).

MADPlaySoundData

This will play a sound data on a MADLibrary driver channel. If you want to play a snd resource, use MADPlaySndHandle function instead of this one.

```
OSErr MADPlaySound( Ptr soundData, long soundDataSize, long channel, long
                    note, long amp, long loopBegin, long loopSize, unsigned
                    long rate);
```

`soundData`	Pointer on raw data
`soundDataSize`	Sound size
`channel`	Channel ID which will be used to play this sound
`note`	note ID, 0xFF : play this sound at normal rate
`amp`	amplitude of this sound, 8 or 16
`loopBegin`	Loop begin
`loopSize`	Loop size

`rate`   sample rate of this sound data

DESCRIPTION

The `MADPlaySound` function is used to play a sound data pointer on a MADLibrary driver channel.