

Musical Libraries for Think C & CodeWarrior

By Antoine ROSSET - ©1992 - 1994 Copyright
Translated by Mike VENTURI

Musical Libraries

The functions of this library offers to the user to play in multitasking (interrupts) a MAD (PlayerPRO) and MODs (PC, Amiga, Atari) type music.

To use the following functions, install the library MAD-Library in your project and make an `#include "MAD.h"`, `#include "RDriver.h"` in your file ".c". The global variables are generally accessible: Music Oscilloscope, tempo, pitch, volume, thePartition, etc.

OSErr RInitMusic(short HardwareType, short Tracks);

See Example.c to find a example of initialization.

Initializes the intern parameters of the library routines. It is strongly advised to launch this routine a the beginning of your program. To be called only ONCE! Needs approximately a free RAM space of a 100k. The value Integer:

short HardwareType

ASCMono = Hardware addressing, needs the ASC
ASCStereo = Hardware addressing, needs the ASC
(PowerMac has no ASC)

SMMono = Sound Manager **3.0**

SMStereo = Sound Manager **3.0**

SMDSP = Sound Manager **3.0 + 16Bit**

short Tracks

Number of tracks, see example.c to have an example....

OSErr RLoadMusic(Str255 Name);

Str255 Name

The name of the music file to load, expressed in Pascal chain. Ex: "pSpace". This music has to be a MAD music, produced by Player PRO (MAD Type).

You have to set the current directory by SetVol (0L, vRefnum); if the file is in another directory.

See example.c to have examples how to load and play MODs musics, instead of MAD music.

After using **RLoadMusic** or **RLoadMusicRsrc**, update the tracks number with this:

```

if( theFileInt->Tracks != DriveTrackNo)
{
    DriveTrackNo = theFileInt->Tracks;
    ChangeTrackNo( DriveTrackNo);
}

```

(see Example.c)

OSErr RLoadMusicRsrc(OSType RsrcType, int RsrcID);

Str255 RsrcType The resource type. By example: MADF when you save a music with Auto-Exec in Player PRO. This music has to be a MAD music, produced by Player PRO (MAD Type).

Str255 RsrcID RsrcID

OSErr RPlayMusic();

Plays the music PREVIOUSLY loaded into memory by the RLoadMusic(). If the function RLoadMusic() has not been called before the risks of a crash are VERY important.

Put the Boolean Reading to true, to start the lecture: **Reading = true.**

OSErr RStopMusic();

Stops the music that is in the course of being played. Corresponds to a pause, if you call RPlayMusic(), the music restarts there where you left it.

OSErr RResetMusic();

Puts back the music at the beginning of the partition.

OSErr RClearMusic();

Empties the memory of the music that is in it now. Use this function before you re-call RLoadMusic to charge the next music.

OSErr RQuitMusic();

Empties the memory used by RInitMusic(), you MUST call this function BEFORE you quit your program.

Error messages sent by these functions:

- 1 Lack of live memory, change it in Set Project.
- 2 Reading error of the music file.
- 3 music file incompatible with the routines of this library.

An example of the utilisation of these functions is supplied with the libraries.

For more informations about globals, function in Driver.h, contact me.

3

Antoine ROSSET
16 BD Tranchées
1206 Geneva
SWITZERLAND

FAX: (+41 22) 346 11 97
BBS: FC ProOnline (+41 22) 738 26 26
Internet: rosset@dial.eunet.ch
Compuserve: 100277,164