

#\$+K!

## MMedia for wxWindows

Guilhem Lavaux

March 2000

### Contents

[Introduction](#)

[MMboard: a sample MMedia application](#)

[Class reference](#)

---

<sup>C</sup>ontents

<sup>C</sup>ontents

<sup>b</sup>rowse00001

<sup>K</sup> Contents

<sup>D</sup>isableButton("Up")

## \$#+K! Introduction

The MMedia wxWindows extension is a wxWindows library which provides you a full set of multimedia classes including sound recording/playing, cd audio playing and video playing. The API is portable and can be used on any supported systems with the insurance the behaviour will be the same.

### File structure

---

lntroduction

topic0

browse00002

K Introduction

DisableButton("Up")

**MMboard: a sample MMedia application**

To be written.

---

<sup>M</sup>Mboard: a sample MMedia application  
<sup>m</sup>mboard  
<sup>b</sup>rowse00004  
<sup>K</sup> MMboard a sample MMedia application  
<sup>D</sup>isableButton("Up")

## Class reference

These are the main Mmedia classes.

[wxCDAudio](#)

[wxCDAudioLinux](#)

[wxCDAudioWin](#)

---

Class reference

classref

rowse00005

Class reference

isableButton("Up")



## **File structure**

These are the files that comprise the mmedia library.

**sndbase.h** Header for wxSoundStream base class and wxSoundFormat base class.

**sndbase.cpp** Basic objects implementation.

**sndfile.h** wxSoundFileStream base class header.

**sndfile.cpp** wxSoundFileStream base class implementation.

**sndpcm.h** wxSoundFormatPcm class header.

**sndpcm.cpp** wxSoundFormatPcm class implementation.

**sndcpcm.h** wxSoundCodecPcm class header (PCM converter).

**sndcpcm.cpp** wxSoundCodecPcm class implementation (PCM converter).

**sndulaw.h**

**sndulaw.cpp**

**sndg72x.h**

**sndg72x.cpp**

**sndoss.h**

**sndoss.cpp**

**sndesd.h**

**sndesd.cpp**

**sndwin.h**

**sndwin.cpp**

**cdbase.h**

**cdbase.cpp**

**cdunix.h**

---

File structure

topic1

browse00003

<sup>K</sup> File structure

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId('mmedia.hlp', `topic0')")

**cdunix.cpp**

**cdwin.h**

**cdwin.cpp**

**vidbase.h**

**vidbase.cpp**

**vidxanm.h**

**vidxanm.cpp**

**vidwin.h**

**vidwin.cpp**

`##+K!`**wxCDAudio**

wxheadingDerived from

wxObject

wxheadingData structures

```
typedef struct wxCDtime {  
    wxUint8 track  
};
```

```
typedef enum    PLAYING, PAUSED, STOPPED    CDstatus
```

---

`w`xCDAudio

`w`xcdaudio

`b`rowse00006

`K` wxCDAudio

`E`nableButton("Up");ChangeButtonBinding("Up", "JumpId(`mmedia.hlp', `classref')")



\$#+K! **wxCDAudioLinux**

wxheadingDerived from

wxCDAudio

wxheadingData structures

wxheadingMembers

wxCDAudioLinux::wxCDAudioLinux

wxCDAudioLinux::~wxCDAudioLinux

wxCDAudioLinux::Play

wxCDAudioLinux::Pause

wxCDAudioLinux::Resume

wxCDAudioLinux::GetStatus

wxCDAudioLinux::GetTime

wxCDAudioLinux::GetToc

wxCDAudioLinux::Ok

wxCDAudioLinux::OpenDevice

---

<sup>w</sup>xCDAudioLinux

<sup>w</sup>xcdaudiolinux

<sup>b</sup>rowse00007

<sup>K</sup> wxCDAudioLinux

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId(`mmedia.hlp', `classref')")

`$#+K!` **wxCDAudioWin**

wxheadingDerived from

wxCDAudio

wxheadingData structures

```
typedef struct CDAW\_Internal {  
    MCIDEVICEID dev\_id  
};
```

wxheadingMembers

wxCDAudioWin::wxCDAudioWin  
wxCDAudioWin::~~wxCDAudioWin  
wxCDAudioWin::Play  
wxCDAudioWin::Pause  
wxCDAudioWin::Resume  
wxCDAudioWin::GetStatus  
wxCDAudioWin::GetTime  
wxCDAudioWin::GetToc  
wxCDAudioWin::Ok

---

<sup>w</sup>xCDAudioWin

<sup>w</sup>xcdaudiowin

<sup>b</sup>rowse00018

<sup>K</sup> wxCDAudioWin

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId('mmedia.hlp', `classref')")



**`$#+K! wxCDAudioLinux::wxCDAudioLinux`**

**`wxCDAudioLinux()`<sup>K</sup>**

**`wxCDAudioLinux(const char* dev_name)`<sup>K</sup>**

---

**`wxCDAudioLinux::wxCDAudioLinux`**

**`wxcdaudiolinuxwxcdaudiolinux`**

**`browse00008`**

**`K wxCDAudioLinux wxCDAudioLinux`**

**`EnableButton("Up");ChangeButtonBinding("Up", "JumpId( mmedia.hlp',  
`wxcdaudiolinux')")`**

**`K wxCDAudioLinux`**

**`K wxCDAudioLinux`**

\$#+K!wxCDAudioLinux::~~wxCDAudioLinux

~wxCDAudioLinux()<sup>K</sup>

---

<sup>w</sup>xCDAudioLinux::~~wxCDAudioLinux

<sup>w</sup>xcdaudiolinuxdtor

<sup>b</sup>rowse00009

<sup>K</sup> wxCDAudioLinux ~wxCDAudioLinux

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId(^mmedia.hlp',  
`xcdaudiolinux')")

<sup>K</sup> ~wxCDAudioLinux

<sup>\$#+K!</sup>**wxCDAudioLinux::Play**

**bool Play(const wxCDtime& *beg\_time*, const wxCDtime& *end\_time*)**<sup>K</sup>

---

<sup>w</sup>xCDAudioLinux::Play

<sup>w</sup>xcdaudiolinuxplay

<sup>b</sup>rowse00010

<sup>K</sup> wxCDAudioLinux Play

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId(^mmedia.hlp',  
`xcdaudiolinux')")

<sup>K</sup> Play

<sup>\$#+K!</sup>**wxCDAudioLinux::Pause**

**bool Pause()**<sup>K</sup>

---

<sup>w</sup>xCDAudioLinux::Pause

<sup>w</sup>xcdaudiolinuxpause

<sup>b</sup>rowse00011

<sup>K</sup> wxCDAudioLinux Pause

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId(^mmedia.hlp',  
`xcdaudiolinux')")

<sup>K</sup> Pause

**`$#+K!wxCDAudioLinux::Resume`**

**`bool Resume()`**<sup>K</sup>

---

<sup>w</sup>`xCDAudioLinux::Resume`

<sup>w</sup>`xcdaudiolinuxresume`

<sup>b</sup>`rowse00012`

<sup>K</sup> `wxCDAudioLinux Resume`

<sup>E</sup>`nableButton("Up");ChangeButtonBinding("Up", "JumpId(^mmedia.hlp',  
`xcdaudiolinux')")`

<sup>K</sup> `Resume`



**wxCDAudioLinux::GetStatus**

**CDstatus GetStatus()**

---

**wxCDAudioLinux::GetStatus**

**wxcdaudiolinuxgetstatus**

**rowse00013**

**wxCDAudioLinux GetStatus**

**enableButton("Up");ChangeButtonBinding("Up", "JumpId(mmedia.hlp',  
`wxcdaudiolinux')")**

**GetStatus**

<sup>\$#+K!</sup>**wxCDAudioLinux::GetTime**

**wxCDtime GetTime()**<sup>K</sup>

---

<sup>w</sup>xCDAudioLinux::GetTime

<sup>w</sup>xcdaudiolinuxgettime

<sup>b</sup>rowse00014

<sup>K</sup> wxCDAudioLinux GetTime

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId(^mmedia.hlp',  
`xcdaudiolinux')")

<sup>K</sup> GetTime

\$#+K!wxCDAudioLinux::GetToc

CDtoc& GetToc()<sup>K</sup>

---

<sup>w</sup>xCDAudioLinux::GetToc

<sup>w</sup>xcdaudiolinuxgettoc

<sup>b</sup>rowse00015

<sup>K</sup> wxCDAudioLinux GetToc

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId(^mmedia.hlp',  
`xcdaudiolinux')")

<sup>K</sup> GetToc

\$#+K! **wxCDAudioLinux::Ok**

constfuncboolOk

---

<sup>w</sup>xCDAudioLinux::Ok

<sup>w</sup>xcdaudiolinuxok

<sup>b</sup>rowse00016

<sup>K</sup> wxCDAudioLinux Ok

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId(`mmedia.hlp',  
`xcdaudiolinux')")

<sup>\$#+K!</sup>**wxCDAudioLinux::OpenDevice**

**void OpenDevice(const char\* *dev\_name*)**<sup>K</sup>

---

<sup>w</sup>xCDAudioLinux::OpenDevice

<sup>w</sup>xcdaudiolinuxopendevic

<sup>b</sup>rowse00017

<sup>K</sup> wxCDAudioLinux OpenDevice

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId(^mmedia.hlp',  
`xcdaudiolinux')")

<sup>K</sup> OpenDevice

\$#+K! **wxCDAudioWin::wxCDAudioWin**

**wxCDAudioWin()**<sup>K</sup>

**wxCDAudioWin(const char\* dev\_name)**<sup>K</sup>

---

<sup>w</sup>**wxCDAudioWin::wxCDAudioWin**

<sup>w</sup>**wxcdaudiowinwxcdaudiowin**

<sup>b</sup>**rowse00019**

<sup>K</sup> **wxCDAudioWin wxCDAudioWin**

<sup>E</sup>**nableButton("Up");ChangeButtonBinding("Up", "JumpId( mmedia.hlp',  
`wxcdaudiowin')")**

<sup>K</sup> **wxCDAudioWin**

<sup>K</sup> **wxCDAudioWin**

**\$#+K!wxCDAudioWin::~~wxCDAudioWin**

**~wxCDAudioWin()<sup>K</sup>**

---

<sup>w</sup>xCDAudioWin::~~wxCDAudioWin

<sup>w</sup>xcdaudiowindtor

<sup>b</sup>rowse00020

<sup>K</sup> wxCDAudioWin ~wxCDAudioWin

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId(^mmedia.hlp',  
`xcdaudiowin')")

<sup>K</sup> ~wxCDAudioWin

<sup>\$#+K!</sup>**wxCDAudioWin::Play**

**bool Play(const wxCDtime& *beg\_time*, const wxCDtime& *end\_time*)**<sup>K</sup>

---

<sup>w</sup>xCDAudioWin::Play

<sup>w</sup>xcdaudiowinplay

<sup>b</sup>rowse00021

<sup>K</sup> wxCDAudioWin Play

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId(\`mmedia.hlp',  
`xcdaudiowin')")

<sup>K</sup> Play



**\$#+K! wxCDAudioWin::Pause**

**bool Pause()**<sup>K</sup>

---

<sup>w</sup>xCDAudioWin::Pause

<sup>w</sup>xcdaudiowinpause

<sup>b</sup>rowse00022

<sup>K</sup> wxCDAudioWin Pause

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId(\mmedia.hlp',

<sup>`</sup>wxcdaudiowin')")

<sup>K</sup> Pause

**`$#+K!wxCDAudioWin::Resume`**

**`bool Resume()`**<sup>K</sup>

---

<sup>w</sup>`xCDAudioWin::Resume`

<sup>w</sup>`xcdaudiowinresume`

<sup>b</sup>`rowse00023`

<sup>K</sup> `wxCDAudioWin Resume`

<sup>E</sup>`nableButton("Up");ChangeButtonBinding("Up", "JumpId(\mmedia.hlp',  
`xcdaudiowin')")`

<sup>K</sup> `Resume`

\$#+K! **wxCDAudioWin::GetStatus**

**CDstatus GetStatus()**<sup>K</sup>

---

wxCDAudioWin::GetStatus

wxcdaudiowingetstatus

browse00024

K wxCDAudioWin GetStatus

EnableButton("Up");ChangeButtonBinding("Up", "JumpId(\mmedia.hlp',  
`wxcdaudiowin')")

K GetStatus

<sup>\$#+K!</sup>**wxCDAudioWin::GetTime**

**wxCDtime GetTime()**<sup>K</sup>

---

<sup>w</sup>xCDAudioWin::GetTime

<sup>w</sup>xcdaudiowingettextime

<sup>b</sup>rowse00025

<sup>K</sup> wxCDAudioWin GetTime

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId(\ mmedia.hlp',  
`xcdaudiowin')")

<sup>K</sup> GetTime

**\$#+K!wxCDAudioWin::GetToc**

**const CDtoc& GetToc()<sup>K</sup>**

---

<sup>w</sup>xCDAudioWin::GetToc

<sup>w</sup>xcdaudiowingettoc

<sup>b</sup>rowse00026

<sup>K</sup> wxCDAudioWin GetToc

<sup>E</sup>nableButton("Up");ChangeButtonBinding("Up", "JumpId(\mmedia.hlp',  
`xcdaudiowin')")

<sup>K</sup> GetToc

`$#+K!wxCDAudioWin::Ok`

`constfuncboolOk`

---

`wxCDAudioWin::Ok`

`wxcdaudiowinok`

`browse00027`

`K wxCDAudioWin Ok`

`EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`mmedia.hlp',  
`xcdaudiowin')")`







