

#\$+K!

MMedia for wxWindows

Guilhem Lavaux

March 2000

Contents

[Introduction](#)

[MMboard: a sample MMedia application](#)

[Class reference](#)

^Contents

^Contents

^browse00001

^K Contents

^DisableButton("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.

MMboard: a sample MMedia application
mboard
rowse00004
K MMboard a sample MMedia application
isableButton("Up")

\$#+K! **Class reference**

These are the main Mmedia classes.

[wxCDAudio](#)

[wxCDAudioLinux](#)

[wxCDAudioWin](#)

^Class reference

^Classref

^browse00005

^K Class reference

^DisableButton("Up")

\$#+K! **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

K File structure

EnableButton("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

^wxCDAudioLinux

^wxcdaudiolinux

^browse00007

^K wxCDAudioLinux

^EnableButton("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

^wxCDAudioWin

^wxcdaudiowin

^browse00018

^K wxCDAudioWin

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

`$#+K! wxCDAudioLinux::wxCDAudioLinux`

`wxCDAudioLinux()`^K

`wxCDAudioLinux(const char* dev_name)`^K

`wxCDAudioLinux::wxCDAudioLinux`

`wxcdaudiolinuxwxcdaudiolinux`

`browse00008`

`K wxCDAudioLinux wxCDAudioLinux`

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

`K wxCDAudioLinux`

`K wxCDAudioLinux`

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

~wxCDAudioLinux()^K

^wxCDAudioLinux::~~wxCDAudioLinux

^wxcdaudiolinuxdtdor

^browse00009

^K wxCDAudioLinux ~wxCDAudioLinux

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

^K ~wxCDAudioLinux

^{\$#+K!}**wxCDAudioLinux::Play**

bool Play(const wxCDtime& *beg_time*, const wxCDtime& *end_time*)^K

^wxCDAudioLinux::Play

^wxcdaudiolinuxplay

^browse00010

^K wxCDAudioLinux Play

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

^K Play

^{\$#+K!}**wxCDAudioLinux::Pause**

bool Pause()^K

^wxCDAudioLinux::Pause

^wxcdaudiolinuxpause

^browse00011

^K wxCDAudioLinux Pause

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

^K Pause

`$#+K!wxCDAudioLinux::Resume`

`bool Resume()`^K

^w`xCDAudioLinux::Resume`

^w`xcdaudiolinuxresume`

^b`rowse00012`

^K `wxCDAudioLinux Resume`

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

^K `Resume`

wxCDAudioLinux::GetStatus

CDstatus GetStatus()

wxCDAudioLinux::GetStatus

wxcdaudiolinuxgetstatus

rowse00013

wxCDAudioLinux GetStatus

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

GetStatus

^{\$#+K!}**wxCDAudioLinux::GetTime**

wxCDtime GetTime()^K

^wxCDAudioLinux::GetTime

^wxcdaudiolinuxgettime

^browse00014

^K wxCDAudioLinux GetTime

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

^K GetTime

\$#+K!wxCDAudioLinux::GetToc

CDtoc& GetToc()^K

^wxCDAudioLinux::GetToc

^wxcdaudiolinuxgettoc

^browse00015

^K wxCDAudioLinux GetToc

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

^K GetToc

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

constfuncboolOk

^wxCDAudioLinux::Ok

^wxcdaudiolinuxok

^browse00016

^K wxCDAudioLinux Ok

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

^{\$#+K!}**wxCDAudioLinux::OpenDevice**

void OpenDevice(const char* *dev_name*)^K

^wxCDAudioLinux::OpenDevice

^wxcdaudiolinuxopendevic

^browse00017

^K wxCDAudioLinux OpenDevice

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

^K OpenDevice

wxCDAudioWin::wxCDAudioWin

wxCDAudioWin()^K

wxCDAudioWin(const char* dev_name)^K

wxCDAudioWin::wxCDAudioWin

wxcdaudiowinwxcdaudiowin

rowse00019

wxCDAudioWin wxCDAudioWin

**nableButton("Up");ChangeButtonBinding("Up", "JumpId(mmedia.hlp',
`wxcdaudiowin')")**

wxCDAudioWin

wxCDAudioWin

^{\$#+K!}**wxCDAudioWin::~~wxCDAudioWin**

~wxCDAudioWin()^K

^w**wxCDAudioWin::~~wxCDAudioWin**

^w**wxcdaudiowindtor**

^b**rowse00020**

^K **wxCDAudioWin ~wxCDAudioWin**

^E**nableButton("Up");ChangeButtonBinding("Up", "JumpId(\ mmedia.hlp',
`wxcdaudiowin')")**

^K **~wxCDAudioWin**

^{\$#+K!}**wxCDAudioWin::Play**

bool Play(const wxCDtime& *beg_time*, const wxCDtime& *end_time*)^K

^wxCDAudioWin::Play

^wxcdaudiowinplay

^browse00021

^K wxCDAudioWin Play

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

^K Play

\$#+K! wxCDAudioWin::Pause

bool Pause()^K

^wxCDAudioWin::Pause

^wxcdaudiowinpause

^browse00022

^K wxCDAudioWin Pause

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

^K Pause

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

bool Resume()^K

wxCDAudioWin::Resume

wxcdaudiowinresume

browse00023

K wxCDAudioWin Resume

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

K Resume

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

CDstatus GetStatus()K

wxCDAudioWin::GetStatus

wxcdaudiowingetstatus

browse00024

K wxCDAudioWin GetStatus

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

K GetStatus

^{\$#+K!}**wxCDAudioWin::GetTime**

wxCDtime GetTime()^K

^wxCDAudioWin::GetTime

^wxcdaudiowingettextime

^browse00025

^K wxCDAudioWin GetTime

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

^K GetTime

\$#+K!wxCDAudioWin::GetToc

const CDtoc& GetToc()^K

^wxCDAudioWin::GetToc

^wxcdaudiowingettoc

^browse00026

^K wxCDAudioWin GetToc

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

^K GetToc

`$#+K!wxCDAudioWin::Ok`

`constfuncboolOk`

`wxCDAudioWin::Ok`

`wxcdaudiowinok`

`browse00027`

`K wxCDAudioWin Ok`

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

