

基于 MFC 的多媒体播放器的实现

刘雅琴¹, 夏玉杰²

(1. 洛阳师范学院信息技术学院, 洛阳 471022; 2. 洛阳师范学院物理与电子信息学院, 洛阳 471022)

摘 要: 在 VC++ 6.0 平台上利用 Windows Media Player 控件实现了在程序中播放一段视频和音频的功能。测试结果证明该方法简单易行, 极大缩短了程序的开发时间。

关键词: MFC; 控件; 消息; ActiveX

Realization of multimedia player based on MFC

LIU Ya-qin¹, XIA Yu-jie²

(1. College of Information Technology, Luoyang Normal University, Luoyang 471022, China;

2. College of Physics and Electronic Information, Luoyang Normal University, Luoyang 471022, China)

Abstract: This paper realizes the functions of playing a section of video and audio using Windows Media Player control on the platform of VC++ 6.0. The test results show that this method is simple and shortens development time of the program greatly.

Key words: MFC; control; message; ActiveX

0 引言

MFC (Microsoft Foundation Class) 是用来编写 Windows 应用程序的 C++ 类集, 其中封装了大部分 Windows API 函数和 Windows 控件, 使用 MFC 类库和 Visual C++ 提供的高度可视的应用程序开发工具, 可使应用程序开发变得简单, 提高代码的可靠性和可重用性^[1-2]。随着多媒体性能的极大提升, 程序中经常要播放一段视频或者一段音频, 如果能够简单地插入音频和视频, 将极大缩短程序的开发时间^[3]。本文基于 MFC 在 Visual C++ 环境下利用 Windows Media Player 控件实现了音频视频文件的播放。

1 Windows Media Player 播放器

在程序中简单地插入视频和音频, 可以采用 Windows Media Player 控件来完成该任务^[4]。

Windows Media Player 是 Windows 系统自带的媒体播放器, 通常简称 WMP, 可以播放 MP3, WMA, WAV 等音频文件, 视频方面可以播放 AVI, MPEG-1, 安装 DVD 解码器以后可以播放 MPEG-2, DVD 等。Windows Media Player 控件将功能封装在多个类中, 每个类负责逻辑上相对独立的一组功能, CWMP-Player4 类负责与 Windows Media Player 控件进行交互。CWMPControls 类负责对媒体播放的控制,

CWMPSettings 负责播放器的基本设置等。下面以 windows media player10 为例列出它的常用类和方法, 见表 1。

2 基于 MFC 的多媒体播放器的实现

本系统利用 windows media player 控件, 在 VC++ 平台上, 实现播放、暂停、继续和停止音频视频文件。其实现步骤为(下面的源码中粗斜体字为添加的代码):

(1) 创建一个项目文件, 名称为 PlayMedia

打开 VisualC++ 6.0, 使用 AppWizard [exe] 建立一个 MFC SDI 的应用程序, 项目文件名称为 PlayMedia。

(2) 激活添加 ActiveX 控件对话框, 并插入 Windows Media Player 控件

在项目创建完成后, 需要通过 Project | Add To Project | Components and Controls 菜单项目激活添加 ActiveX 控件对话框, 然后选择 Windows Media Player, 点击 Insert 按钮向项目中插入该控件。该版本的控件名为“Windows Media Player”, 相应的控件文件变为 wmp.dll。

收稿日期: 2008-07-14

作者简介: 刘雅琴(1969-), 女, 副教授, 硕士, 1994 年本科毕业于河南师范大学, 2004 年在西安电子科技大学获取硕士学位, 主要研究方向为多媒体技术。

表 1 Windows Media Player10 常用的类与方法

类	方法	说明
CWMPLayer4	GetCurrentMedia ()	获得当前的媒体对象
	GetControls	返回一个 CWMPLayer4 类的对象
	GetCurrentPlayList ()	获取当前播放列表对象
	GetSettings ()	返回一个 CWMPLayer4 类的对象
	GetUrl ()	获得多媒体文件的 URL
	SetUrl ()	设置多媒体文件的 URL
CWMPLayerSettings	GetAutoStart ()	获得自动播放信息
	GetMute ()	获得静音信息
	GetPlayCount ()	获得播放的次数
	GetRate ()	获得播放的速度
	GetVolume ()	获得音量
	SetAutoStart ()	设置自动播放
	SetPlay ()	设置播放
	SetRate ()	设置播放速度
	SetVolume ()	设置音量
CWMPLayerControls	GetCurrentPosition ()	当前进度
	next ()	下一曲
	pause ()	暂停
	play ()	播放
	previous ()	上一曲
	stop ()	停止
	fastForward ()	快进
	fastReverse ()	快退

(3) 设计 CPlayMediaView 类

为 CPlayMediaView 类添加 3 个成员——一个用来播放多媒体文件,一个用来保存待播放的文件名,另一个记录暂停状态,修改后的 CPlayMediaView. H 代码如下:

```
class CWMPLayer4; //类的声明
class CMy12 - 3View :public CView
{
protected://create from serialization only
    CMy12 - 3View ();
    DECLARE - DYNCREATE(CMy12 - 3View)
// Attributes
public :
    CMy12 - 3Doc * GetDocument ();
    CWMPLayer4 *m - Video;//用来播放视频
    CWMPLayer4 *m - Music;//用来播放音频
    CString m - strVideo;//视频文件名
    CString m - strMusic;//音频文件名
```

```
BOOL m - bPaused; //记录暂停状态的变量
// Operations
```

```
.....
}
```

(4) 初始化和释放指针,在 PlayMediaView. Cpp 文件添加代码

```
# include "wmplayer4. h "
# include "wmpcontrols. h "
# include "wmpsettings. h "
CPlayMediaView::CPlayMediaView ()
{ // TODO: add construction code here
    m - Video = new CWMPLayer4 ;
//初始化视频指针
    m - Music = new CWMPLayer4 ;
//初始化音频指针
    m - bPaused = FALSE;
//初始化暂停变量
}
```

```
CPlayMediaView::~CPlayMediaView ()
{ delete m - Video ;//释放视频指针
  delete m - Music ;//释放音频指针
}
```

(5)使用 Create 函数初始化 ActiveX 控件

响应 View 类中的 WM - CREATE 消息,创建对应于视频和音频播放的两个控件。添加的代码为:

```
int CPlayMediaView:: OnCreate ( LPCREATESTRUCT lpCreateStruct)
{ .....
    HCURSOR hCursor = ::LoadCursor (NULL ,
    IDC - ARROW) ;
    m - Video Create (AfxRegisterWndClass (CS - HREDRAW |CS - VREDRAW |CS - DBLCLKS |CS - PARENTDC ,hCursor ,0 ,0) ,NULL ,WS - VISIBLE |WS - CHILD ,CRect (0 ,0 ,0 ,0) ,this ,0) ;
    m - Video GetSettings (). SetMute (TRUE) ;
    m - Music Create (AfxRegisterWndClass (CS - HREDRAW |CS - VREDRAW |CS - DBLCLKS |CS - PARENTDC ,hCursor ,0 ,0) ,
    NULL ,WS - VISIBLE |WS - CHILD ,CRect (0 ,0 ,0 ,0) ,this ,0) ;
    m - Music ShowWindow (SW - HIDE) ;
    return 0 ;
}
```

(6) 菜单设计

为了操作播放过程,还要添加“操作”及相关菜单项^[5],如表2所示。

表2 菜单项及其属性

ID	Caption	Separator
	操作 (&O)	
ID - OPER - OPENV	打开视频	
ID - OPER - PLAYV	播放视频	
ID - OPER - PAUSEV	暂停/继续	
ID - OPER - STOPV	停止视频	
ID - OPER - OPENA	打开音频	
ID - OPER - PLAYA	播放音频	
ID - OPER - PAUSEA	暂停/继续音频	
ID - OPER - STOPA	停止音频	

然后通过 Class Wizard 为 CPlayMediaView 类添加响应的菜单响应。下面主要说一下视频播放各消息响应函数(音频播放与此大同小异)。

//打开视频代码:

```
void CPlayMediaView::OnOperOperv()  
{// TODO: Add your command handler code here  
    static char BASED_CODE szFilter[] =  
        "Video Files ( *.avi; *.mpg; *.mpeg ) |  
        *.avi; *.mpg; *.mpeg |";  
    CFileDialog fileDlg ( TRUE, NULL, NULL,  
        OFN - HIDEREADONLY, szFilter );  
    if (fileDlg. DoModal () == IDOK)  
        m - strVideo = fileDlg. GetPathName ();}
```

//播放视频

```
void CPlayMediaView::OnOperPlayv()  
{// TODO: Add your command handler code here  
    m - Video GetControls(). stop ();  
    m - Video GetSettings (). SetAutoStart  
(TRUE);  
    m - Video SetUrl (m - strVideo);}
```

//暂停/继续视频

```
void CPlayMediaView::OnOperPausev()  
{  
    //TODO: Add your command handler code here  
    if (m - bPaused)  
        {m - bPaused = FALSE;
```

```
m - Video GetControls(). play ();}  
else  
{m - bPaused = TRUE;  
    m - Video GetControls(). pause ();}  
}
```

//停止视频

```
void CPlayMediaView::OnOperStopv()  
{m - Video GetControls (). stop ();}
```

(7) 为播放视频的控件设置窗口尺寸

有了以上代码,程序还不能够完全正常运行,需要为播放视频的控件设置窗口尺寸。为此需要响应 View 的 WM - SIZE 消息,该消息响应函数如下:

```
void CMy12 - 3View::OnSize (UINT nType, int cx,  
int cy)  
{ CView::OnSize (nType, cx, cy);  
    //TODO: Add your message handler code here  
    CView::OnSize (nType, cx, cy);  
    CRect rect;  
    GetClientRect (rect);  
    m - pPlayer MoveWindow (rect);}
```

(8) 编译、连接运行即可实现要求的功能,播放器运行界面略。

3 结束语

介绍了 VC++ 环境下插入 Windows Media Player 的方法,探讨了多媒体指针的定义、初始化和释放的方法,研究了 ActiveX 控件的创建、菜单选项的设置及响应控件窗口尺寸的设置等。实现了在 Visual C++ 环境下利用 Windows Media Player 控件实现音频视频文件的播放。本方法简单易行,方便用户在所编程序中添加音频视频文件。

参考文献:

- [1] 黄维通,等. Visual C++ 面向对象与可视化程序设计[M]. 北京:清华大学出版社,2003.
- [2] 唐俊明. Visual C++ 6.0 编程实例与技巧[M]. 北京:高等教育出版社,2002.
- [3] 齐维毅,申海,等. API 函数在制作多媒体播放器中的应用[J]. 沈阳师范大学,2006,24(2):199 - 201.
- [4] 李博轩. Visual C++ 6.0 多媒体开发指南[M]. 北京:清华大学出版社,2000.
- [5] 孙鑫,余安萍. VC++ 深入详解[M]. 北京:电子工业出版社,2006.

责任编辑:么丽苹