原 Media Player Classic - HC 源代码分析 6：MediaInfo选项卡 （CPPageFileMediaInfo）

2013年10月30日 00:42:45　阅读数：6084

===

Media Player Classic - HC 源代码分析系列文章列表：

Media Player Classic - HC 源代码分析 1：整体结构

Media Player Classic - HC 源代码分析 2：核心类 （CMainFrame） （1）

Media Player Classic - HC 源代码分析 3：核心类 （CMainFrame） （2）

Media Player Classic - HC 源代码分析 4：核心类 （CMainFrame） （3）

Media Player Classic - HC 源代码分析 5：关于对话框 （CAboutDlg）

Media Player Classic - HC 源代码分析 6：MediaInfo选项卡 （CPPageFileMediaInfo）

Media Player Classic - HC 源代码分析 7：详细信息选项卡（CPPageFileInfoDetails）

===

前几篇文章分析了Media Player Classic - HC（mpc-hc）的核心类（CMainFrame）：

Media Player Classic - HC 源代码分析 2：核心类 （CMainFrame） （1）

Media Player Classic - HC 源代码分析 3：核心类 （CMainFrame） （2）

Media Player Classic - HC 源代码分析 4：核心类 （CMainFrame） （3）

核心类分析完之后，分析了一下CAboutDlg：

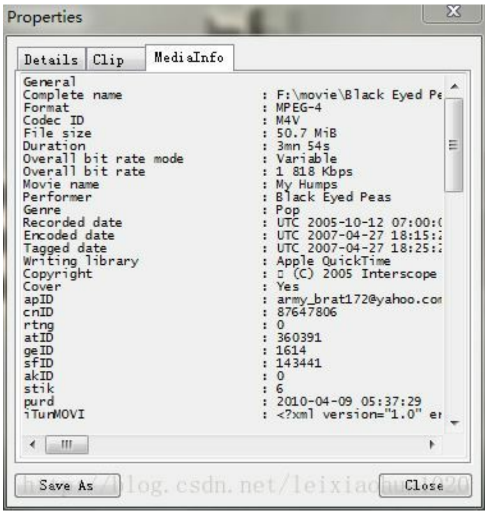
Media Player Classic - HC 源代码分析 5：关于对话框 （CAboutDlg）

发现CAboutDlg和普通的MFC对话框类其实没有什么区别。CAboutDlg功能相对比较简单，本文将会分析一个功能相对比较复杂的类：MediaInfo选项卡。在播放视频的时候，右键点击视频->选择“属性”->MediaInfo就可以查看该选项卡。一般情况下，该选项卡给出了正在播放的视频文件的详细参数（确实是非常的详细），包括：封装格式，视频编码，音频编码等等。是获取视频详细参数的最佳途径。
该选项卡的功能实际上是调用了开源项目MediaInfo的库。MediaInfo之前已经进行过详细介绍：

C++中使用MediaInfo库获取视频信息

MediaInfo使用简介（新版本支持HEVC）

在此不再重复。先看看该选项卡长什么样子。



```
General
Complete name                            : F:\movie\Black Eyed Pe
Format                                   : MPEG-4
Codec ID                                 : M4V
File size                                : 50.7 MiB
Duration                                 : 3mn 54s
Overall bit rate mode                    : Variable
Overall bit rate                         : 1 818 Kbps
Movie name                               : My Humps
Performer                                : Black Eyed Peas
Genre                                    : Pop
Recorded date                            : UTC 2005-10-12 07:00:(
Encoded date                             : UTC 2007-04-27 18:15:2
Tagged date                              : UTC 2007-04-27 18:25:2
Writing library                          : Apple QuickTime
Copyright                                : □ (C) 2005 Interscope
Cover                                    : Yes
apID                                     : army_brat172@yahoo.com
cnID                                     : 87647806
rtng                                     : 0
atID                                     : 360391
geID                                     : 1614
sfID                                     : 143441
akID                                     : 0
stik                                     : 6
purd                                     : 2010-04-09 05:37:29
iTunMOVI                                 : <?xml version="1.0" en
```

先来看看MediaInfo选项卡类的定义是什么样的吧。该类的定义位于PPageFileMediaInfo.h文件中。

[cpp]

```cpp
/* 雷霄骅
 * 中国传媒大学/数字电视技术
 * leixiaohua1020@126.com
 *
 */
/*
 * (C) 2009-2013 see Authors.txt
 *
 * This file is part of MPC-HC.
 *
 * MPC-HC is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 3 of the License, or
 * (at your option) any later version.
 *
 * MPC-HC is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program.  If not, see <http://www.gnu.org/licenses/>.
 *
 */

#pragma once

// CPPageFileMediaInfo dialog
// 【属性】页面里面的【MediaInfo】
class CPPageFileMediaInfo : public CPropertyPage
{
    DECLARE_DYNAMIC(CPPageFileMediaInfo)

private:
    CComPtr<IFilterGraph> m_pFG;
public:
    //构造函数都是两个参数
    CPPageFileMediaInfo(CString fn, IFilterGraph* pFG);
    virtual ~CPPageFileMediaInfo();

    // Dialog Data
    enum { IDD = IDD_FILEMEDIAINFO };
    //显示信息的控件
    CEdit m_mediainfo;
    CString m_fn;
    CFont* m_pCFont;
    //信息
    CString MI_Text;

#if !USE_STATIC_MEDIAINFO
    static bool HasMediaInfo();
#endif
protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
    //初始化，加载MediaInfo库，读取文件信息
    virtual BOOL OnInitDialog();

    DECLARE_MESSAGE_MAP()

public:
    //显示窗口，并不做其他事情
    afx_msg void OnShowWindow(BOOL bShow, UINT nStatus);
};
```

该类和普通的MFC对话框类差别也不大。需要注意的有以下几点：

1.有一个变量:CComPtr<IFilterGraph>   m_pFG,这个是mpc-hc中的变量,先不分析该变量的全部代码,在这里仅说一下它的作用:获取正在播放的视频文件的路径。

2.有一个控件类：CEdit m_mediainfo，对应界面上那个大框框，用于显示信息。

3.有一个字符串变量：CString MI_Text，用于存储MediaInfo得到的媒体信息。

下面来看看具体类的实现，该类的实现位于PPageFileMediaInfo.cpp文件中。

[cpp]

```cpp
/* 雷霄骅
 * 中国传媒大学/数字电视技术
 * leixiaohua1020@126.com
 *
 */
/*
 * (C) 2009-2013 see Authors.txt
 *
 * This file is part of MPC-HC.
 *
 * MPC-HC is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 3 of the License, or
 * (at your option) any later version.
 *
 * MPC-HC is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program.  If not, see <http://www.gnu.org/licenses/>.
 *
 */

// PPageFileMediaInfo.cpp : implementation file


#include "stdafx.h"
#include "mplayerc.h"
#include "PPageFileMediaInfo.h"
#include "WinAPIUtils.h"

#if USE_STATIC_MEDIAINFO
#include "MediaInfo/MediaInfo.h"
using namespace MediaInfoLib;
#else
#include "MediaInfoDLL.h"
using namespace MediaInfoDLL;
#endif


// CPPageFileMediaInfo dialog

IMPLEMENT_DYNAMIC(CPPageFileMediaInfo, CPropertyPage)
CPPageFileMediaInfo::CPPageFileMediaInfo(CString fn, IFilterGraph* pFG)
    : CPropertyPage(CPPageFileMediaInfo::IDD, CPPageFileMediaInfo::IDD)
    , m_fn(fn)
    , m_pFG(pFG)
    , m_pCFont(nullptr)
{
}

CPPageFileMediaInfo::~CPPageFileMediaInfo()
{
    delete m_pCFont;
    m_pCFont = nullptr;
}

void CPPageFileMediaInfo::DoDataExchange(CDataExchange* pDX)
{
    __super::DoDataExchange(pDX);
    DDX_Control(pDX, IDC_MIEDIT, m_mediainfo);
}


BEGIN_MESSAGE_MAP(CPPageFileMediaInfo, CPropertyPage)
    ON_WM_SHOWWINDOW()
END_MESSAGE_MAP()

// CPPageFileMediaInfo message handlers
static WNDPROC OldControlProc;

static LRESULT CALLBACK ControlProc(HWND control, UINT message, WPARAM wParam, LPARAM lParam)
{
    if (message == WM_KEYDOWN) {
        if ((LOWORD(wParam) == 'A' || LOWORD(wParam) == 'a')
                && (GetKeyState(VK_CONTROL) < 0)) {
            CEdit* pEdit = (CEdit*)CWnd::FromHandle(control);
            pEdit->SetSel(0, pEdit->GetWindowTextLength(), TRUE);
            return 0;
        }
    }

    return CallWindowProc(OldControlProc, control, message, wParam, lParam); // call edit control's own windowproc
}
//初始化，加载MediaInfo库，读取文件信息
BOOL CPPageFileMediaInfo::OnInitDialog()
{
    __super::OnInitDialog();

    if (!m_pCFont) {
        m_pCFont = DEBUG_NEW CFont;
    }
    if (!m_pCFont) {
        return TRUE;
    }

    if (m_fn.IsEmpty()) {
        BeginEnumFilters(m_pFG, pEF, pBF) {
            CComQIPtr<IFileSourceFilter> pFSF = pBF;
            if (pFSF) {
                //当前文件路径
                LPOLESTR pFN = nullptr;
                //媒体类型
                AM_MEDIA_TYPE mt;
                //获取当前文件的路径和媒体类型
                if (SUCCEEDED(pFSF->GetCurFile(&pFN, &mt)) && pFN && *pFN) {
                    m_fn = CStringW(pFN);
                    CoTaskMemFree(pFN);
                }
                break;
            }
        }
        EndEnumFilters;
    }

#if USE_STATIC_MEDIAINFO
    //使用静态库MediaInfo
    //文件路径
    MediaInfoLib::String f_name = m_fn;
    MediaInfoLib::MediaInfo MI;
#else
    MediaInfoDLL::String f_name = m_fn;
    MediaInfo MI;
#endif
    //设置
    MI.Option(_T("ParseSpeed"), _T("0"));
    MI.Open(f_name);
    MI.Option(_T("Complete"));
    MI.Option(_T("Language"), _T("  Config_Text_ColumnSize;30"));
    //信息字符串
    MI_Text = MI.Inform().c_str();
    MI.Close();
    if (!MI_Text.Find(_T("Unable to load"))) {
        MI_Text = _T("");
    }

    LOGFONT lf;
    ZeroMemory(&lf, sizeof(lf));
    lf.lfPitchAndFamily = DEFAULT_PITCH | FF_MODERN;
    // The empty string will fallback to the first font that matches the other specified attributes.
    LPCTSTR fonts[] = { _T("Lucida Console"), _T("Courier New"), _T("") };
    // Use a negative value to match the character height instead of the cell height.
    int fonts_size[] = { -10, -11, -11 };
    UINT i = 0;
    BOOL success;
    do {
        _tcscpy_s(lf.lfFaceName, fonts[i]);
        lf.lfHeight = fonts_size[i];
        success = IsFontInstalled(fonts[i]) && m_pCFont->CreateFontIndirect(&lf);
        i++;
    } while (!success && i < _countof(fonts));
    //控件设置字体和内容
    m_mediainfo.SetFont(m_pCFont);
    m_mediainfo.SetWindowText(MI_Text);

    // subclass the edit control
    OldControlProc = (WNDPROC)SetWindowLongPtr(m_mediainfo.m_hWnd, GWLP_WNDPROC, (LONG_PTR)ControlProc);

    return TRUE;  // return TRUE unless you set the focus to a control
    // EXCEPTION: OCX Property Pages should return FALSE
}
//显示or不显示？
void CPPageFileMediaInfo::OnShowWindow(BOOL bShow, UINT nStatus)
{
    __super::OnShowWindow(bShow, nStatus);
    __super::OnShowWindow(bShow, nStatus);
    if (bShow) {
        GetParent()->GetDlgItem(IDC_BUTTON_MI)->ShowWindow(SW_SHOW);
    } else {
        GetParent()->GetDlgItem(IDC_BUTTON_MI)->ShowWindow(SW_HIDE);
    }
}

#if !USE_STATIC_MEDIAINFO
bool CPPageFileMediaInfo::HasMediaInfo()
{
    MediaInfo MI;
    return MI.IsReady();
}
#endif
```

可以看出，主要的工作都是在OnInitDialog()函数中实现的。大体的步骤如下：

1.通过调用pFSF->GetCurFile(&pFN, &mt)，获得当前文件的路径，存入pFN中。

2.因为字符串类型不同，几经转换把pFN转换为MediaInfo可以识别的字符串f_name

3.根据该路径，调用MediaInfo库，获得视频的详细信息存入字符串变量MI_Text。

4.将MI_Text显示到控件上。

总体说来，过程并不复杂，理解起来还是比较简单的。

文章标签： mpc-hc 源代码 directshow 开源 播放器

个人分类： MediaInfo MPC-HC

所属专栏： 开源多媒体项目源代码分析