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

==

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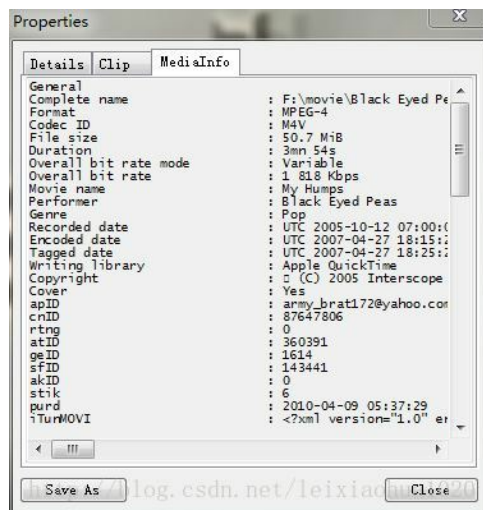
先来看看MediaInfo选项卡类的定义是什么样的吧。该类的定义位于PPageFileMediaInfo.h文件中。

```cpp
1.  /* 雷霄骅
2.   * 中国传媒大学/数字电视技术
3.   * leixiaohua1020@126.com
4.   *
5.   */
6.  /*
7.   * (C) 2009-2013 see Authors.txt
8.   *
9.   * This file is part of MPC-HC.
10.  *
11.  * MPC-HC is free software; you can redistribute it and/or modify
12.  * it under the terms of the GNU General Public License as published by
13.  * the Free Software Foundation; either version 3 of the License, or
14.  * (at your option) any later version.
15.  *
16.  * MPC-HC is distributed in the hope that it will be useful,
17.  * but WITHOUT ANY WARRANTY; without even the implied warranty of
18.  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
19.  * GNU General Public License for more details.
20.  *
21.  * You should have received a copy of the GNU General Public License
22.  * along with this program.  If not, see <http://www.gnu.org/licenses/>.
23.  *
24.  */
25.
26.  #pragma once
27.
28.  // CPPageFileMediaInfo dialog
29.  // 【属性】页面里面的【MediaInfo】
30.  class CPPageFileMediaInfo : public CPropertyPage
31.  {
32.      DECLARE_DYNAMIC(CPPageFileMediaInfo)
33.
34.  private:
35.      CComPtr<IFilterGraph> m_pFG;
36.  public:
37.      //构造函数都是两个参数
38.      CPPageFileMediaInfo(CString fn, IFilterGraph* pFG);
39.      virtual ~CPPageFileMediaInfo();
40.
41.      // Dialog Data
42.      enum { IDD = IDD_FILEMEDIAINFO };
43.      //显示信息的控件
44.      CEdit m_mediainfo;
45.      CString m_fn;
46.      CFont* m_pCFont;
47.      //信息
48.      CString MI_Text;
49.
50.  #if !USE_STATIC_MEDIAINFO
51.      static bool HasMediaInfo();
52.  #endif
53.  protected:
54.      virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
55.      //初始化，加载MediaInfo库，读取文件信息
56.      virtual BOOL OnInitDialog();
57.
58.      DECLARE_MESSAGE_MAP()
59.
60.  public:
61.      //显示窗口，并不做其他事情
62.      afx_msg void OnShowWindow(BOOL bShow, UINT nStatus);
63.  };
```

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
/* 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)
{
```

```cpp
76.          if (message == WM_KEYDOWN) {
77.              if ((LOWORD(wParam) == 'A' || LOWORD(wParam) == 'a')
78.                      && (GetKeyState(VK_CONTROL) < 0)) {
79.                  CEdit* pEdit = (CEdit*)CWnd::FromHandle(control);
80.                  pEdit->SetSel(0, pEdit->GetWindowTextLength(), TRUE);
81.                  return 0;
82.              }
83.          }
84.
85.          return CallWindowProc(OldControlProc, control, message, wParam, lParam); // call edit control's own windowproc
86.      }
87.      //初始化，加载MediaInfo库，读取文件信息
88.      BOOL CPPageFileMediaInfo::OnInitDialog()
89.      {
90.          __super::OnInitDialog();
91.
92.          if (!m_pCFont) {
93.              m_pCFont = DEBUG_NEW CFont;
94.          }
95.          if (!m_pCFont) {
96.              return TRUE;
97.          }
98.
99.          if (m_fn.IsEmpty()) {
100.             BeginEnumFilters(m_pFG, pEF, pBF) {
101.                 CComQIPtr<IFileSourceFilter> pFSF = pBF;
102.                 if (pFSF) {
103.                     //当前文件路径
104.                     LPOLESTR pFN = nullptr;
105.                     //媒体类型
106.                     AM_MEDIA_TYPE mt;
107.                     //获取当前文件的路径和媒体类型
108.                     if (SUCCEEDED(pFSF->GetCurFile(&pFN, &mt)) && pFN && *pFN) {
109.                         m_fn = CStringW(pFN);
110.                         CoTaskMemFree(pFN);
111.                     }
112.                     break;
113.                 }
114.             }
115.             EndEnumFilters;
116.         }
117.
118.     #if USE_STATIC_MEDIAINFO
119.         //使用静态库MediaInfo
120.         //文件路径
121.         MediaInfoLib::String f_name = m_fn;
122.         MediaInfoLib::MediaInfo MI;
123.     #else
124.         MediaInfoDLL::String f_name = m_fn;
125.         MediaInfo MI;
126.     #endif
127.         //设置
128.         MI.Option(_T("ParseSpeed"), _T("0"));
129.         MI.Open(f_name);
130.         MI.Option(_T("Complete"));
131.         MI.Option(_T("Language"), _T("  Config_Text_ColumnSize;30"));
132.         //信息字符串
133.         MI_Text = MI.Inform().c_str();
134.         MI.Close();
135.         if (!MI_Text.Find(_T("Unable to load"))) {
136.             MI_Text = _T("");
137.         }
138.
139.         LOGFONT lf;
140.         ZeroMemory(&lf, sizeof(lf));
141.         lf.lfPitchAndFamily = DEFAULT_PITCH | FF_MODERN;
142.         // The empty string will fallback to the first font that matches the other specified attributes.
143.         LPCTSTR fonts[] = { _T("Lucida Console"), _T("Courier New"), _T("") };
144.         // Use a negative value to match the character height instead of the cell height.
145.         int fonts_size[] = { -10, -11, -11 };
146.         UINT i = 0;
147.         BOOL success;
148.         do {
149.             _tcscpy_s(lf.lfFaceName, fonts[i]);
150.             lf.lfHeight = fonts_size[i];
151.             success = IsFontInstalled(fonts[i]) && m_pCFont->CreateFontIndirect(&lf);
152.             i++;
153.         } while (!success && i < _countof(fonts));
154.         //控件设置字体和内容
155.         m_mediainfo.SetFont(m_pCFont);
156.         m_mediainfo.SetWindowText(MI_Text);
157.
158.         // subclass the edit control
159.         OldControlProc = (WNDPROC)SetWindowLongPtr(m_mediainfo.m_hWnd, GWLP_WNDPROC, (LONG_PTR)ControlProc);
160.
161.         return TRUE;  // return TRUE unless you set the focus to a control
162.         // EXCEPTION: OCX Property Pages should return FALSE
163.     }
164.     //显示or不显示？
165.     void CPPageFileMediaInfo::OnShowWindow(BOOL bShow, UINT nStatus)
166.     {
167.         __super::OnShowWindow(bShow, nStatus);
```

```
167.        __super::OnShowWindow(bShow, nStatus);
168.        if (bShow) {
169.            GetParent()->GetDlgItem(IDC_BUTTON_MI)->ShowWindow(SW_SHOW);
170.        } else {
171.            GetParent()->GetDlgItem(IDC_BUTTON_MI)->ShowWindow(SW_HIDE);
172.        }
173.    }
174.
175.    #if !USE_STATIC_MEDIAINFO
176.    bool CPPageFileMediaInfo::HasMediaInfo()
177.    {
178.        MediaInfo MI;
179.        return MI.IsReady();
180.    }
181.    #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