

ffds show 源代码分析 9：编解码器有关类的总结

2013年11月13日 00:33:44 阅读数：6514

ffds show源代码分析系列文章列表：

[ffds show 源代码分析 1：整体结构](#)

[ffds show 源代码分析 2：位图覆盖滤镜（对话框部分Dialog）](#)

[ffds show 源代码分析 3：位图覆盖滤镜（设置部分Settings）](#)

[ffds show 源代码分析 4：位图覆盖滤镜（滤镜部分Filter）](#)

[ffds show 源代码分析 5：位图覆盖滤镜（总结）](#)

[ffds show 源代码分析 6：对解码器的dll的封装（libavcodec）](#)

[ffds show 源代码分析 7：libavcodec视频解码器类（TvideoCodecLibavcodec）](#)

[ffds show 源代码分析 8：视频解码器类（TvideoCodecDec）](#)

[ffds show 源代码分析 9：编解码器有关类的总结](#)



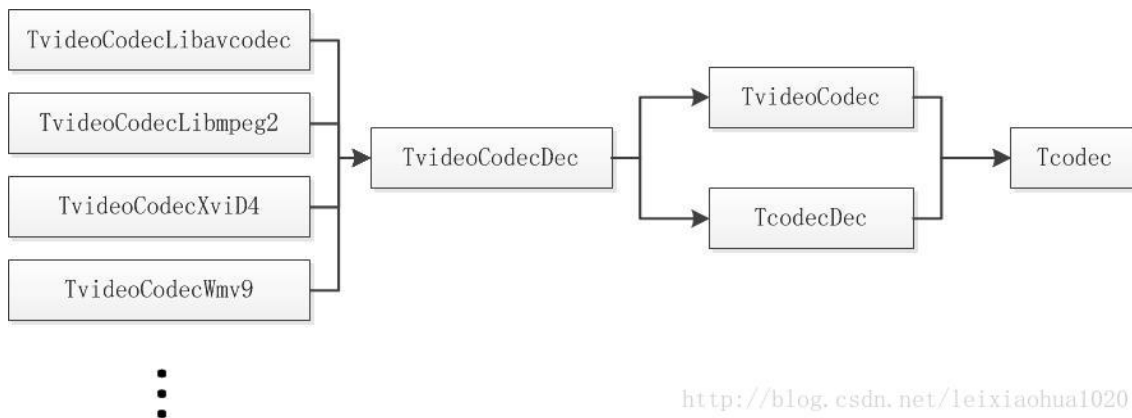
前几篇文章已经完成了ffds show解码器封装的大部分代码的分析：

[ffds show 源代码分析 6：对解码器的dll的封装（libavcodec）](#)

[ffds show 源代码分析 7：libavcodec视频解码器类（TvideoCodecLibavcodec）](#)

[ffds show 源代码分析 8：视频解码器类（TvideoCodecDec）](#)

本文再做最后一点的分析。在ffds show中有如下继承关系：



<http://blog.csdn.net/leixiaohua1020>

前文已经分析过TvideoCodecLibavcodec，TvideoCodecDec，在这里我们看一下他们的父类：TvideoCodec，TcodecDec，以及前两个类的父类Tcodec。

其实本文介绍的这3个类充当了接口的作用，TvideoCodecDec继承TvideoCodec，TcodecDec，以及这两个类继承Tcodec，都使用了virtual的方式。

先来看看TvideoCodec。注意这个类强调的是【视频】：

```

1. //编解码器的父类
2. class TvideoCodec : virtual public Tcodec
3. {
4. public:
5.     TvideoCodec(IffdshowBase *Ideci);
6.     virtual ~TvideoCodec();
7.     bool ok;
8.     int connectedSplitter;
9.     bool isInterlacedRawVideo;
10.    Rational containerSar;
11.
12.    struct CAPS {
13.        enum {
14.            NONE = 0,
15.            VIS_MV = 1,
16.            VIS_QUANTS = 2
17.        };
18.    };
19.
20.    virtual void end(void) {}
21. };

```

可以看出TvideoCodec定义非常的简单，只包含了视频编解码器会用到的一些变量。注意，是编解码器，不仅仅是解码器。

再来看看TcodecDec。注意这个类强调的是【解码】：

```

1. //实现了解码器的祖父类
2. class TcodecDec : virtual public Tcodec
3. {
4. private:
5.     IdecSink *sink;
6. protected:
7.     comptrQ<IffdshowDec> deciD;
8.     TcodecDec(IffdshowBase *Ideci, IdecSink *Isink);
9.     virtual ~TcodecDec();
10.    virtual HRESULT flushDec(void) {
11.        return S_OK;
12.    }
13. public:
14.     virtual HRESULT flush(void);
15. };

```

可以看出TcodecDec定义非常简单，只包含了解码器需要的一些变量，注意不限于视频解码器，还包含音频解码器。有两个变量比较重要：

```

IdecSink *sink;
comptrQ<IffdshowDec> deciD;

```

最后来看一下Tcodec。这个类不再继承任何类：

```

1. //编解码器的祖父类，都是虚函数
2. class Tcodec
3. {
4. protected:
5.     const Tconfig *config;
6.     comptr<IffdshowBase> deci;
7.     Tcodec(IffdshowBase *Ideci);
8.     virtual ~Tcodec();
9. public:
10.     AVCodecID codecId;
11.     virtual int getType(void) const = 0;
12.     virtual const char_t* getName(void) const {
13.         return getMovieSourceName(getType());
14.     }
15.     virtual void getEncoderInfo(char_t *buf, size_t buflen) const {
16.         ff_strncpy(buf, _l("unknown"), buflen);
17.         buf[buflen - 1] = '\0';
18.     }
19.     static const char_t* getMovieSourceName(int source);
20.
21.     virtual HRESULT flush() {
22.         return S_OK;
23.     }
24.     virtual HRESULT BeginFlush() {
25.         return S_OK;
26.     }
27.     virtual HRESULT EndFlush() {
28.         return S_OK;
29.     }
30.     virtual bool onSeek(REFERENCE_TIME segmentStart) {
31.         return false;
32.     }
33. };

```

可以看出，该类定义了一些编解码器会用到的公共函数。有几个变量还是比较重要的：

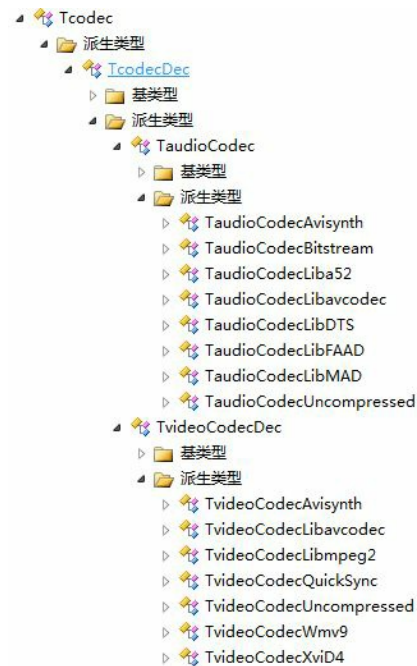
```

const Tconfig *config;
comptr<IffdshowBase> deci;
Tcodec(IffdshowBase *Ideci);
AVCodecID codecId

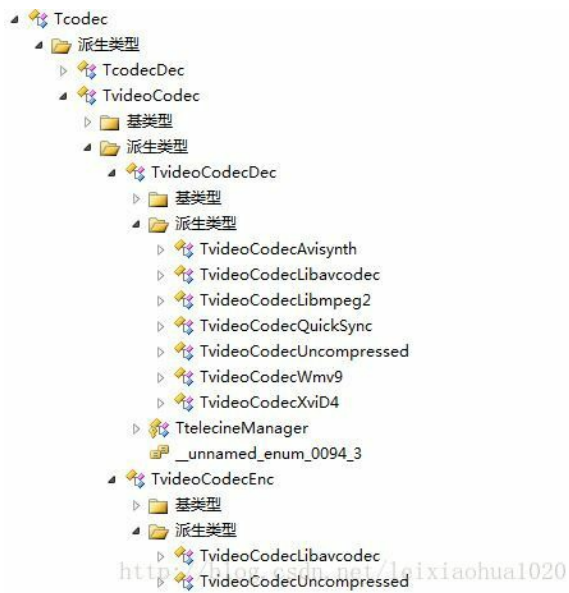
```

自此，我们可以总结出ffdshow编解码器这部分继承关系如下（图太大了，截成两张）：

从TcodecDec继承下来的如下图所示。包含视频解码器以及音频解码器。



从TvideoCodec继承下来的如下图所示。包含了解码器类和编码器类。



总算大体上完成了，关于ffdshow解码器封装的内容就先告一段落吧。

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/leixiaohua1020/article/details/15493961>

文章标签： [ffdshow](#) [编码器](#) [解码器](#) [源代码](#) [视频](#)

个人分类： [ffdshow](#)

所属专栏： [开源多媒体项目源代码分析](#)

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com