原 ffdshow 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

==========

ffdshow源代码分析系列文章列表：

ffdshow 源代码分析 1： 整体结构

ffdshow 源代码分析 2： 位图覆盖滤镜（对话框部分Dialog）

ffdshow 源代码分析 3： 位图覆盖滤镜（设置部分Settings）

ffdshow 源代码分析 4： 位图覆盖滤镜（滤镜部分Filter）

ffdshow 源代码分析 5： 位图覆盖滤镜（总结）

ffdshow 源代码分析 6： 对解码器的dll的封装（libavcodec）

ffdshow 源代码分析 7： libavcodec视频解码器类（TvideoCodecLibavcodec）

ffdshow 源代码分析 8： 视频解码器类（TvideoCodecDec）

ffdshow 源代码分析 9： 编解码器有关类的总结

==========

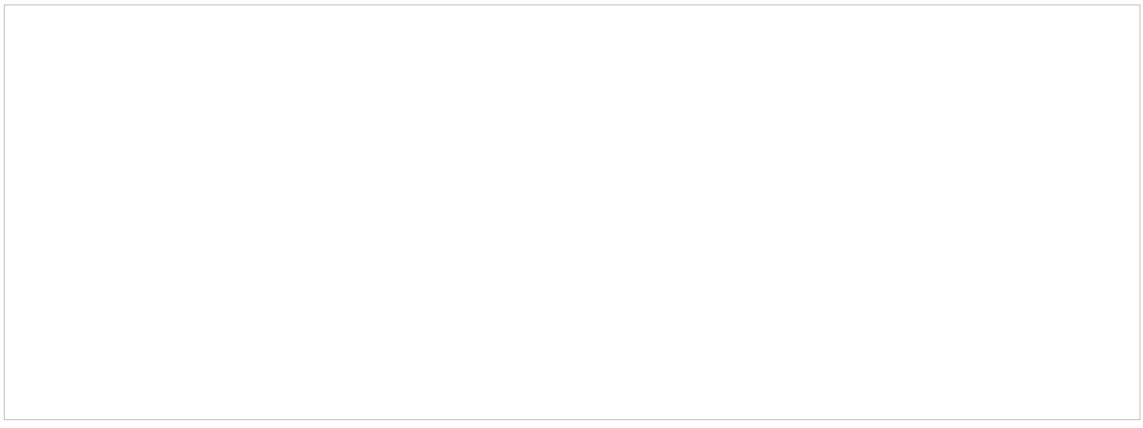
前几篇文章已经完成了ffdshow解码器封装的大部分代码的分析：

ffdshow 源代码分析 6： 对解码器的dll的封装（libavcodec）

ffdshow 源代码分析 7： libavcodec视频解码器类（TvideoCodecLibavcodec）

ffdshow 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ffdshow中有如下继承关系：



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】：

```cpp
//编解码器的父类
class TvideoCodec : virtual public Tcodec
{
public:
    TvideoCodec(IffdshowBase *Ideci);
    virtual ~TvideoCodec();
    bool ok;
    int connectedSplitter;
    bool isInterlacedRawVideo;
    Rational containerSar;

    struct CAPS {
        enum {
            NONE = 0,
            VIS_MV = 1,
            VIS_QUANTS = 2
        };
    };

    virtual void end(void) {}
};
```

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】：

```cpp
//实现了解码器的祖父类
class TcodecDec : virtual public Tcodec
{
private:
    IdecSink *sink;
protected:
    comptrQ<IffdshowDec> deciD;
    TcodecDec(IffdshowBase *Ideci, IdecSink *Isink);
    virtual ~TcodecDec();
    virtual HRESULT flushDec(void) {
        return S_OK;
    }
public:
    virtual HRESULT flush(void);
};
```

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：

```cpp
//编解码器的祖父类，都是虚函数
class Tcodec
{
protected:
    const Tconfig *config;
    comptr<IffdshowBase> deci;
    Tcodec(IffdshowBase *Ideci);
    virtual ~Tcodec();
public:
    AVCodecID codecId;
    virtual int getType(void) const = 0;
    virtual const char_t* getName(void) const {
        return getMovieSourceName(getType());
    }
    virtual void getEncoderInfo(char_t *buf, size_t buflen) const {
        ff_strncpy(buf, _l("unknown"), buflen);
        buf[buflen - 1] = '\0';
    }
    static const char_t* getMovieSourceName(int source);

    virtual HRESULT flush() {
        return S_OK;
    }
    virtual HRESULT BeginFlush() {
        return S_OK;
    }
    virtual HRESULT EndFlush() {
        return S_OK;
    }
    virtual bool onSeek(REFERENCE_TIME segmentStart) {
        return false;
    }
};
```

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。

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