原 MediaInfo源代码分析 4：Inform()函数

2013年10月08日 21:40:09 阅读数：4006

==========

MediaInfo源代码分析系列文章列表：

==========

我们来看一下MediaInfo中的Inform()函数的内部调用过程

首先Inform()函数封装了MediaInfo_Internal类中的Inform()函数

```cpp
//返回文件信息
String MediaInfo::Inform(size_t)
{
    //封装了一层
    return Internal->Inform();
}
```

查看一下MediaInfo_Internal类中的Inform()函数的源代码：

```cpp
// 获取信息
Ztring MediaInfo_Internal::Inform()
{
    CS.Enter();
    if (Info && Info->Status[File__Analyze::IsUpdated])
        Info->Open_Buffer_Update();
    CS.Leave();

    if (MediaInfoLib::Config.Inform_Get()==__T("MPEG-7"))
        return Export_Mpeg7().Transform(*this);
    if (MediaInfoLib::Config.Inform_Get()==__T("PBCore") || MediaInfoLib::Config.Inform_Get()==__T("PBCore_1.2"))
        return Export_PBCore().Transform(*this);
    if (MediaInfoLib::Config.Inform_Get()==__T("reVTMD"))
        return __T("reVTMD is disabled due to its non-free licensing."); //return Export_reVTMD().Transform(*this);
    //获取相应的信息
    if (!(
        MediaInfoLib::Config.Inform_Get(__T("General")).empty()
     && MediaInfoLib::Config.Inform_Get(__T("Video")).empty()
     && MediaInfoLib::Config.Inform_Get(__T("Audio")).empty()
     && MediaInfoLib::Config.Inform_Get(__T("Text")).empty()
     && MediaInfoLib::Config.Inform_Get(__T("Chapters")).empty()
     && MediaInfoLib::Config.Inform_Get(__T("Image")).empty()
     && MediaInfoLib::Config.Inform_Get(__T("Menu")).empty()
     ))
    {
        //获取各种信息
        //Retour即为返回的字符串
        Ztring Retour;
        Retour+=MediaInfoLib::Config.Inform_Get(__T("File_Begin"));
        Retour+=MediaInfoLib::Config.Inform_Get(__T("General_Begin"));
        Retour+=Inform(Stream_General, 0, false);
        Retour+=MediaInfoLib::Config.Inform_Get(__T("General_End"));
        if (Count_Get(Stream_Video))
            Retour+=MediaInfoLib::Config.Inform_Get(__T("Video_Begin"));
        for (size_t I1=0; I1<Count_Get(Stream_Video); I1++)
        {
            Retour+=Inform(Stream_Video, I1, false);
            if (I1!=Count_Get(Stream_Video)-1)
                Retour+=MediaInfoLib::Config.Inform_Get(__T("Video_Middle"));
        }
        if (Count_Get(Stream_Video))
            Retour+=MediaInfoLib::Config.Inform_Get(__T("Video_End"));
        if (Count_Get(Stream_Audio))
            Retour+=MediaInfoLib::Config.Inform_Get(__T("Audio_Begin"));
        for (size_t I1=0; I1<Count_Get(Stream_Audio); I1++)
        {
            Retour+=Inform(Stream_Audio, I1, false);
            if (I1!=Count_Get(Stream_Audio)-1)
```

```
            if (I1!=Count_Get(Stream_Audio)-1)
                Retour+=MediaInfoLib::Config.Inform_Get(__T("Audio_Middle"));
        }
        if (Count_Get(Stream_Audio))
            Retour+=MediaInfoLib::Config.Inform_Get(__T("Audio_End"));
        if (Count_Get(Stream_Text))
            Retour+=MediaInfoLib::Config.Inform_Get(__T("Text_Begin"));
        for (size_t I1=0; I1<Count_Get(Stream_Text); I1++)
        {
            Retour+=Inform(Stream_Text, I1, false);
            if (I1!=Count_Get(Stream_Text)-1)
                Retour+=MediaInfoLib::Config.Inform_Get(__T("Text_Middle"));
        }
        if (Count_Get(Stream_Text))
            Retour+=MediaInfoLib::Config.Inform_Get(__T("Text_End"));
        if (Count_Get(Stream_Other))
            Retour+=MediaInfoLib::Config.Inform_Get(__T("Chapters_Begin"));
        for (size_t I1=0; I1<Count_Get(Stream_Other); I1++)
        {
            Retour+=Inform(Stream_Other, I1, false);
            if (I1!=Count_Get(Stream_Other)-1)
                Retour+=MediaInfoLib::Config.Inform_Get(__T("Chapters_Middle"));
        }
        if (Count_Get(Stream_Other))
            Retour+=MediaInfoLib::Config.Inform_Get(__T("Chapters_End"));
        if (Count_Get(Stream_Image))
            Retour+=MediaInfoLib::Config.Inform_Get(__T("Image_Begin"));
        for (size_t I1=0; I1<Count_Get(Stream_Image); I1++)
        {
            Retour+=Inform(Stream_Image, I1, false);
            if (I1!=Count_Get(Stream_Image)-1)
                Retour+=MediaInfoLib::Config.Inform_Get(__T("Image_Middle"));
        }
        if (Count_Get(Stream_Image))
            Retour+=MediaInfoLib::Config.Inform_Get(__T("Image_End"));
        if (Count_Get(Stream_Menu))
            Retour+=MediaInfoLib::Config.Inform_Get(__T("Menu_Begin"));
        for (size_t I1=0; I1<Count_Get(Stream_Menu); I1++)
        {
            Retour+=Inform(Stream_Menu, I1, false);
            if (I1!=Count_Get(Stream_Menu)-1)
                Retour+=MediaInfoLib::Config.Inform_Get(__T("Menu_Middle"));
        }
        if (Count_Get(Stream_Menu))
            Retour+=MediaInfoLib::Config.Inform_Get(__T("Menu_End"));
        Retour+=MediaInfoLib::Config.Inform_Get(__T("File_End"));
        //可以在此加入视频质量检测----------------------------------------

        //字符串替换？各种换行符统统改为“\n”---------------------------
        Retour.FindAndReplace(__T("\\r\\n"), __T("\n"), 0, Ztring_Recursive);
        Retour.FindAndReplace(__T("\\r"), __T("\n"), 0, Ztring_Recursive);
        Retour.FindAndReplace(__T("\\n"), __T("\n"), 0, Ztring_Recursive);
        Retour.FindAndReplace(__T("\r\n"), __T("\n"), 0, Ztring_Recursive);
        Retour.FindAndReplace(__T("\r"), __T("\n"), 0, Ztring_Recursive);
        Retour.FindAndReplace(__T("\n"), MediaInfoLib::Config.LineSeparator_Get(), 0, Ztring_Recursive);

        //Special characters
        Retour.FindAndReplace(__T("|SC1|"), __T("\\"), 0, Ztring_Recursive);
        Retour.FindAndReplace(__T("|SC2|"), __T("["), 0, Ztring_Recursive);
        Retour.FindAndReplace(__T("|SC3|"), __T("]"), 0, Ztring_Recursive);
        Retour.FindAndReplace(__T("|SC4|"), __T(","), 0, Ztring_Recursive);
        Retour.FindAndReplace(__T("|SC5|"), __T(";"), 0, Ztring_Recursive);
        Retour.FindAndReplace(__T("|SC6|"), __T("("), 0, Ztring_Recursive);
        Retour.FindAndReplace(__T("|SC7|"), __T(")"), 0, Ztring_Recursive);
        Retour.FindAndReplace(__T("|SC8|"), __T(")"), 0, Ztring_Recursive);
        Retour.FindAndReplace(__T("|SC9|"), __T("),"), 0, Ztring_Recursive);

        return Retour;
    }

    //Informations
    Ztring Retour;
    bool HTML=false;
    bool XML=false;
    bool CSV=false;
    //获取配置信息（输出格式）
    if (MediaInfoLib::Config.Inform_Get()==__T("HTML"))
        HTML=true;
    if (MediaInfoLib::Config.Inform_Get()==__T("XML"))
        XML=true;
    if (MediaInfoLib::Config.Inform_Get()==__T("CSV"))
        CSV=true;

    if (HTML) Retour+=__T("<html>\n\n<head>\n<META http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8\" /></head>\n<body
>\n");
    if (XML)  Retour+=__T("<File>\n");

    for (size_t StreamKind=(size_t)Stream_General; StreamKind<Stream_Max; StreamKind++)
    {
        //Pour chaque type de flux
        for (size_t StreamPos=0; StreamPos<(size_t)Count_Get((stream_t)StreamKind); StreamPos++)
        {
```

```
            //Pour chaque stream
            //输出为HTML
            if (HTML) Retour+=__T("
<table width=\"100%\" border=\"0\" cellpadding=\"1\" cellspacing=\"2\" style=\"border:1px solid Navy\">\n<tr>\n    <td width=\"150\">
<h2>");
            //输出为XML
            if (XML) Retour+=__T("<track type=\"");
            Ztring A=Get((stream_t)StreamKind, StreamPos, __T("StreamKind/String"));
            Ztring B=Get((stream_t)StreamKind, StreamPos, __T("StreamKindPos"));
            if (!XML && !B.empty())
            {
                if (CSV)
                    A+=__T(",");
                else
                    A+=MediaInfoLib::Config.Language_Get(__T("  Config_Text_NumberTag"));
                A+=B;
            }
            Retour+=A;
            if (XML)
            {
                Retour+=__T("\"");
                if (!B.empty())
                {
                    Retour+=__T(" streamid=\"");
                    Retour+=B;
                    Retour+=__T("\"");
                }
            }
            //输出为HTML
            if (HTML) Retour+=__T("</h2></td>\n  </tr>");
            //输出为XML
            if (XML) Retour+=__T(">");
            Retour+=MediaInfoLib::Config.LineSeparator_Get();
            Retour+=Inform((stream_t)StreamKind, StreamPos, false);
            Retour.FindAndReplace(__T("\\"), __T("|SC1|"), 0, Ztring_Recursive);
            if (HTML) Retour+=__T("</table>\n<br />");
            if (XML) Retour+=__T("</track>\n");
            Retour+=MediaInfoLib::Config.LineSeparator_Get();
        }
    }
    //输出为HTML
    if (HTML) Retour+=__T("\n</body>\n</html>\n");
    //输出为XML
    if (XML)  Retour+=__T("</File>\n");
    //字符串替换？
    Retour.FindAndReplace(__T("\\r\\n"), __T("\n"), 0, Ztring_Recursive);
    Retour.FindAndReplace(__T("\\r"), __T("\n"), 0, Ztring_Recursive);
    Retour.FindAndReplace(__T("\\n"), __T("\n"), 0, Ztring_Recursive);
    Retour.FindAndReplace(__T("\r\n"), __T("\n"), 0, Ztring_Recursive);
    Retour.FindAndReplace(__T("\r"), __T("\n"), 0, Ztring_Recursive);
    Retour.FindAndReplace(__T("\n"), MediaInfoLib::Config.LineSeparator_Get(), 0, Ztring_Recursive);

    //Special characters
    Retour.FindAndReplace(__T("|SC1|"), __T("\\"), 0, Ztring_Recursive);
    Retour.FindAndReplace(__T("|SC2|"), __T("["), 0, Ztring_Recursive);
    Retour.FindAndReplace(__T("|SC3|"), __T("]"), 0, Ztring_Recursive);
    Retour.FindAndReplace(__T("|SC4|"), __T(","), 0, Ztring_Recursive);
    Retour.FindAndReplace(__T("|SC5|"), __T(";"), 0, Ztring_Recursive);
    Retour.FindAndReplace(__T("|SC6|"), __T("("), 0, Ztring_Recursive);
    Retour.FindAndReplace(__T("|SC7|"), __T(")"), 0, Ztring_Recursive);
    Retour.FindAndReplace(__T("|SC8|"), __T(")"), 0, Ztring_Recursive);
    Retour.FindAndReplace(__T("|SC9|"), __T("),"), 0, Ztring_Recursive);
    return Retour;
}
```

函数比较复杂，从代码中我们可以看出，Inform()的实质还是使用Get()一个一个取出所有的属性值。

当指定输出为XML或者是HTML的时候，在输出的字符串上加上相应的标签（例如，输出为HTML的时候，字符串每一行上加上“</tr><tr>”，首尾加上“<table></table>”）

具体每一块代码的含义已经写在注释中了。

文章标签： 源代码 MediaInfo Inform

个人分类： MediaInfo

所属专栏： 开源多媒体项目源代码分析