

MediaInfo源代码分析 3：Open()函数

2013年10月08日 20:27:26 阅读数：5422

MediaInfo源代码分析系列文章列表：

[MediaInfo源代码分析 1：整体结构](#)

[MediaInfo源代码分析 2：API函数](#)

[MediaInfo源代码分析 3：Open\(\)函数](#)

[MediaInfo源代码分析 4：Inform\(\)函数](#)

[MediaInfo源代码分析 5：JPEG解析代码分析](#)

我们来看一下MediaInfo中的Open()函数的内部调用过程

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

```
[cpp]
1. //打开文件
2. size_t MediaInfo::Open(const String &File_Name_)
3. {
4.     //封装了一层
5.     return Internal->Open(File_Name_);
6. }
```

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

```
[cpp]
1. //打开文件
2. size_t MediaInfo_Internal::Open(const String &File_Name_)
3. {
4.     Close();
5.
6.     CS.Enter();
7.     MEDIAINFO_DEBUG_CONFIG_TEXT(Debug+=__T("Open, File=");Debug+=Ztring(File_Name_).c_str());
8.     //清空文件列表
9.     Config.File_Names.clear();
10.    if (Config.File_FileNameFormat_Get()==__T("CSV"))
11.    {
12.        Config.File_Names.Separator_Set(0, __T(","));
13.        Config.File_Names.Write(File_Name_);
14.    }
15.    //添加文件名到文件名列表
16.    else if (!File_Name_.empty())
17.        Config.File_Names.push_back(File_Name_);
18.    if (Config.File_Names.empty())
19.    {
20.        CS.Leave();
21.        return 0;
22.    }
23.    Config.File_Names_Pos=1;
24.    CS.Leave();
25.
26.    //Parsing
27.    //解析
28.    if (BlockMethod==1)
29.    {
30.        if (!IsInThread) //If already created, the routine will read the new files
31.        {
32.            Run();
33.            IsInThread=true;
34.        }
35.        return 0;
36.    }
37.    else
38.    {
39.        Entry(); //Normal parsing
40.        return Count_Get(Stream_General);
41.    }
42. }
```

在打开之前首先调用Close()进行清理，将文件路径添加到一个列表（实际上是一个Vector）中，调用Entry()进行解析。

查看一下Entry()函数的源代码：

```
[cpp]
1. //解析函数
2. void MediaInfo_Internal::Entry()
3. {
4.     CS.Enter();
5.     MEDIAINFO_DEBUG_CONFIG_TEXT(Debug+=__T("Entry");)
6.     Config.State_Set(0);
7.     CS.Leave();
8.     //解析文件名 (是否包含MMS?)
9.     //需要libmms
10.    if ((Config.File_Names[0].size())>=6
11.        && Config.File_Names[0][0]==__T('m')
12.        && Config.File_Names[0][1]==__T('m')
13.        && Config.File_Names[0][2]==__T('s')
14.        && Config.File_Names[0][3]==__T(':')
15.        && Config.File_Names[0][4]==__T('/')
16.        && Config.File_Names[0][5]==__T('/'))
17.    || (Config.File_Names[0].size())>=7
18.        && Config.File_Names[0][0]==__T('m')
19.        && Config.File_Names[0][1]==__T('m')
20.        && Config.File_Names[0][2]==__T('s')
21.        && Config.File_Names[0][3]==__T('h')
22.        && Config.File_Names[0][4]==__T(':')
23.        && Config.File_Names[0][5]==__T('/')
24.        && Config.File_Names[0][6]==__T('/'))
25.    #if defined(MEDIAINFO_LIBMMS_YES)
26.        Reader_libmms().Format_Test(this, Config.File_Names[0]);
27.    #else //MEDIAINFO_LIBMMS_YES
28.    {
29.        #if MEDIAINFO_EVENTS
30.            struct MediaInfo_Event_Log_0 Event;
31.            Event.EventCode=MediaInfo_EventCode_Create(MediaInfo_Parser_None, MediaInfo_Event_Log, 0);
32.            Event.Type=0xC0;
33.            Event.Severity=0xFF;
34.            Event.MessageCode=0;
35.            Event.MessageStringU=L"Libmms cupport is disabled due to compilation options";
36.            Event.MessageStringA="Libmms support is disabled due to compilation options";
37.            MediaInfoLib::Config.Event_Send((const int8u*)&Event, sizeof(MediaInfo_Event_Log_0));
38.        #endif //MEDIAINFO_EVENTS
39.    }
40.    #endif //MEDIAINFO_LIBMMS_YES
41.    //是否包含网络协议? (标记为://)
42.    //需要使用libcurl - the multiprotocol file transfer library
43.    else if (Config.File_Names[0].find(__T("://"))!=string::npos)
44.    #if defined(MEDIAINFO_LIBCURL_YES)
45.    {
46.        CS.Enter();
47.        if (Reader)
48.        {
49.            CS.Leave();
50.            return; //There is a problem
51.        }
52.        Reader=new Reader_libcurl();
53.        CS.Leave();
54.
55.        Reader->Format_Test(this, Config.File_Names[0]);
56.
57.        #if MEDIAINFO_NEXTPACKET
58.            if (Config.NextPacket_Get())
59.                return;
60.        #endif //MEDIAINFO_NEXTPACKET
61.    }
62.    #else //MEDIAINFO_LIBCURL_YES
63.    {
64.        #if MEDIAINFO_EVENTS
65.            struct MediaInfo_Event_Log_0 Event;
66.            Event.EventCode=MediaInfo_EventCode_Create(MediaInfo_Parser_None, MediaInfo_Event_Log, 0);
67.            Event.Type=0xC0;
68.            Event.Severity=0xFF;
69.            Event.MessageCode=0;
70.            Event.MessageStringU=L"Libcurl support is disabled due to compilation options";
71.            Event.MessageStringA="Libcurl support is disabled due to compilation options";
72.            MediaInfoLib::Config.Event_Send((const int8u*)&Event, sizeof(MediaInfo_Event_Log_0));
73.        #endif //MEDIAINFO_EVENTS
74.    }
75.    #endif //MEDIAINFO_LIBCURL_YES
76.
77.    #if defined(MEDIAINFO_DIRECTORY_YES)
78.        else if (Dir::Exists(Config.File_Names[0]))
79.            Reader_Directory().Format_Test(this, Config.File_Names[0]);
80.    #endif //MEDIAINFO_DIRECTORY_YES
81.    //是普通文件, 并且文件存在
82.    #if defined(MEDIAINFO_FILE_YES)
83.        else if (File::Exists(Config.File_Names[0]))
84.        {
85.            #if defined(MEDIAINFO_REFERENCES_YES)
86.                string Dwx;
87.                if (Config.File_CheckSideCarFiles_Get() && !Config.File_IsReferenced_Get())
```

```

88.     {
89.         //ZenLib中竟然包含专用于文件名操作的类！
90.         FileName Test(Config.File_Names[0]);
91.         //获得小写的文件名后缀
92.         Ztring FileExtension=Test.Extension_Get();
93.         FileExtension.MakeLowerCase();
94.         //逐个比较（这些都是文本文件的后缀）
95.         if (FileExtension!=__T("dfxp"))
96.         {
97.             Test.Extension_Set(__T("dfxp"));
98.             if (File::Exists(Test))
99.                 Dwx+=" <clip file=\""+Test.Name_Get().To_UTF8()+"\".dfxp\" />\r\n";
100.        }
101.        if (FileExtension!=__T("sami"))
102.        {
103.            Test.Extension_Set(__T("sami"));
104.            if (File::Exists(Test))
105.                Dwx+=" <clip file=\""+Test.Name_Get().To_UTF8()+"\".sami\" />\r\n";
106.        }
107.        if (FileExtension!=__T("sc2"))
108.        {
109.            Test.Extension_Set(__T("sc2"));
110.            if (File::Exists(Test))
111.                Dwx+=" <clip file=\""+Test.Name_Get().To_UTF8()+"\".sc2\" />\r\n";
112.        }
113.        if (FileExtension!=__T("scc"))
114.        {
115.            Test.Extension_Set(__T("scc"));
116.            if (File::Exists(Test))
117.                Dwx+=" <clip file=\""+Test.Name_Get().To_UTF8()+"\".scc\" />\r\n";
118.        }
119.        if (FileExtension!=__T("smi"))
120.        {
121.            Test.Extension_Set(__T("smi"));
122.            if (File::Exists(Test))
123.                Dwx+=" <clip file=\""+Test.Name_Get().To_UTF8()+"\".smi\" />\r\n";
124.        }
125.        //字幕
126.        if (FileExtension!=__T("srt"))
127.        {
128.            Test.Extension_Set(__T("srt"));
129.            if (File::Exists(Test))
130.                Dwx+=" <clip file=\""+Test.Name_Get().To_UTF8()+"\".srt\" />\r\n";
131.        }
132.        if (FileExtension!=__T("stl"))
133.        {
134.            Test.Extension_Set(__T("stl"));
135.            if (File::Exists(Test))
136.                Dwx+=" <clip file=\""+Test.Name_Get().To_UTF8()+"\".stl\" />\r\n";
137.        }
138.
139.        Ztring Name=Test.Name_Get();
140.        Ztring BaseName=Name.SubString(Ztring(), __T("_"));
141.        if (!BaseName.empty())
142.        {
143.            ZtringList List;
144.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.audio.mp4"), Dir::Include_Files);
145.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.sub.dfxp"), Dir::Include_Files);
146.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.sub.sami"), Dir::Include_Files);
147.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.sub.sc2"), Dir::Include_Files);
148.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.sub.scc"), Dir::Include_Files);
149.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.sub.smi"), Dir::Include_Files);
150.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.sub.srt"), Dir::Include_Files);
151.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.sub.stl"), Dir::Include_Files);
152.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.forcesub.dfxp"), Dir::Include_Files);
153.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.forcesub.sami"), Dir::Include_Files);
154.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.forcesub.sc2"), Dir::Include_Files);
155.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.forcesub.scc"), Dir::Include_Files);
156.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.forcesub.smi"), Dir::Include_Files);
157.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.forcesub.srt"), Dir::Include_Files);
158.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.forcesub.stl"), Dir::Include_Files);
159.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.cc.dfxp"), Dir::Include_Files);
160.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.cc.sami"), Dir::Include_Files);
161.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.cc.sc2"), Dir::Include_Files);
162.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.cc.scc"), Dir::Include_Files);
163.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.cc.smi"), Dir::Include_Files);
164.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.cc.srt"), Dir::Include_Files);
165.            List+=Dir::GetAllFileNames(Test.Path_Get()+PathSeparator+BaseName+__T("_*.cc.stl"), Dir::Include_Files);
166.            for (size_t Pos=0; Pos<List.size(); Pos++)
167.                Dwx+=" <clip file=\""+List[Pos].To_UTF8()+"\" />\r\n";
168.        }
169.
170.        if (!Dwx.empty())
171.        {

```

```

172.         Dwx.insert(0, "<?xml version=\"1.0\" encoding=\"UTF-8\" ?>\r\n"
173.             "<indexFile xmlns=\"urn:digimetrics-xml-wrapper\" xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-insta
nce\" xsi:schemaLocation=\"urn:digimetrics-xml-wrapper DMSCLIP.XSD\">\r\n"
174.             "    <clip source=\"main\" file=\""+FileName(Config.File_Names[0]).Name_Get().To_UTF8()+"."+FileName(Co
nfig.File_Names[0]).Extension_Get().To_UTF8()+"\" />\r\n");
175.         Dwx.append("</indexFile>\r\n");
176.         Config.File_FileNameFormat_Set(_T("Dxw"));
177.     }
178. }
179. //为空
180. if (Dwx.empty())
181. {
182.     CS.Enter();
183.     if (Reader)
184.     {
185.         CS.Leave();
186.         return; //There is a problem
187.     }
188.     Reader=new Reader_File();
189.     CS.Leave();
190.     //测试格式
191.     Reader->Format_Test(this, Config.File_Names[0]);
192. }
193. else
194. {
195.     //初始化Buffer
196.     Open_Buffer_Init(Dwx.size(), FileName(Config.File_Names[0]).Path_Get()+PathSeparator+FileName(Config.File_Names[0]
Name_Get());
197.     Open_Buffer_Continue((const int8u*)Dwx.c_str(), Dwx.size());
198.     Open_Buffer_Finalize();
199. }
200. #else //defined(MEDIAINFO_REFERENCES_YES)
201.     CS.Enter();
202.     if (Reader)
203.     {
204.         CS.Leave();
205.         return; //There is a problem
206.     }
207.     Reader=new Reader_File();
208.     CS.Leave();
209.
210.     Reader->Format_Test(this, Config.File_Names[0]);
211. #endif //defined(MEDIAINFO_REFERENCES_YES)
212.
213. #if MEDIAINFO_NEXTPACKET
214.     if (Config.NextPacket_Get())
215.         return;
216. #endif //MEDIAINFO_NEXTPACKET
217. }
218. #endif //MEDIAINFO_FILE_YES
219.
220. CS.Enter();
221. Config.State_Set(1);
222. CS.Leave();
223. }

```

我们可以看出，MediaInfo支持流媒体协议作为输入源（例如mms），但前提是要使用相应的库（libmms，或者libcurl）。它会首先查找文件路径中有没有“://”，判断是否是流媒体输入。

大多数情况下，我们使用本地文件路径作为输入源。这是我们就进入了代码中“#if defined(MEDIAINFO_FILE_YES)”部分下面的代码。具体每块代码的左右都写在代码注释中了，不再细说。最后使用 Reader_File的Format_Test()测试输入文件的格式。

看一下Reader_File的Format_Test()的源代码：

```

1. //用于测试文件格式
2. size_t Reader_File::Format_Test(MediaInfo_Internal* MI, String File_Name)
3. {
4.     //std::cout<<Ztring(File_Name).To_Local().c_str()<<std::endl;
5.
6.     //With Parser MultipleParsing
7.     /*
8.     MI->Open_Buffer_Init((int64u)-1, File_Name);
9.     if (Format_Test_PerParser(MI, File_Name))
10.         return 1;
11.     return 0; //There is a problem
12.     */
13.
14.     //Get the Extension
15.     //文件后缀
16.     Ztring Extension=FileName::Extension_Get(File_Name);
17.     Extension.MakeLowerCase();
18.
19.     //Search the theorical format from extension
20.     InfoMap &FormatList=MediaInfoLib::Config.Format_Get();
21.     InfoMap::iterator Format=FormatList.end();
22.     if (!MI->Config.File_ForceParser_Get().empty())
23.         Format=FormatList.find(MI->Config.File_ForceParser_Get());
24.     if (Format==FormatList.end())
25.     {
26.         Format=FormatList.begin();
27.         while (Format!=FormatList.end())
28.         {
29.             const Ztring &Extensions=FormatList.Get(Format->first, InfoFormat_Extensions);
30.             if (Extensions.find(Extension)!=Error)
31.             {
32.                 if(Extension.size()==Extensions.size())
33.                     break; //Only one extension in the list
34.                 if(Extensions.find(Extension+__T(" "))!=Error
35.                    || Extensions.find(__T(" ") +Extension)!=Error)
36.                     break;
37.             }
38.             ++Format;
39.         }
40.     }
41.     if (Format!=FormatList.end())
42.     {
43.         const Ztring &Parser=Format->second(InfoFormat_Parser);
44.         if (MI->SelectFromExtension(Parser))
45.         {
46.             //Test the theorical format
47.             if (Format_Test_PerParser(MI, File_Name)>0)
48.                 return 1;
49.         }
50.     }
51.     //遍历所有支持格式，测试格式
52.     size_t ToReturn=MI->ListFormats(File_Name);
53.     return ToReturn;
54. }

```

可以看出，最后遍历了所有支持的格式，试图找出合适的格式。

最后我们看一下ListFormats()函数，用于找出匹配的格式。

由代码可见，遍历测试每一种支持的格式。

```

1. int MediaInfo_Internal::ListFormats(const String &File_Name)
2. {
3.     // Multiple
4.     // 看看这些格式是否支持
5.     //从上往下遍历
6.     //如果遇到某一格式，返回值为1，便不再进行
7.     //否则就进行下一个，直到遍历完
8.     #if defined(MEDIAINFO_BDAV_YES)
9.         delete Info; Info=new File_MpegTs(); ((File_MpegTs*)Info)->BDASize=4; if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return 1;
10.    #endif
11.    #if defined(MEDIAINFO_BDMV_YES)
12.        delete Info; Info=new File_Bdmv(); if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return 1;
13.    #endif
14.    #if defined(MEDIAINFO_CDXA_YES)
15.        delete Info; Info=new File_Cdxa(); if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return 1;
16.    #endif
17.    #if defined(MEDIAINFO_DCP_YES)
18.        delete Info; Info=new File_Dcp(); if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return 1;
19.    #endif
20.    #if defined(MEDIAINFO_DVDIF_YES)

```

```

21.         delete Info; Info=new File_DvDif();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
22.     #endif
23.     #if defined(MEDIAINFO_DVDV_YES)
24.         delete Info; Info=new File_Dvdv();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
25.     #endif
26.     #if defined(MEDIAINFO_DXW_YES)
27.         delete Info; Info=new File_Dxw();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
28.     #endif
29.     #if defined(MEDIAINFO_FLV_YES)
30.         //检测
31.         delete Info; Info=new File_Flv();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
32.     #endif
33.     #if defined(MEDIAINFO_GXF_YES)
34.         delete Info; Info=new File_Gxf();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
35.     #endif
36.     #if defined(MEDIAINFO_HLS_YES)
37.         delete Info; Info=new File_Hls();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
38.     #endif
39.     #if defined(MEDIAINFO_IBI_YES)
40.         delete Info; Info=new File_Ibi();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retu
rn 1;
41.     #endif
42.     #if defined(MEDIAINFO_ISM_YES)
43.         delete Info; Info=new File_Ism();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
44.     #endif
45.     #if defined(MEDIAINFO_IVF_YES)
46.         delete Info; Info=new File_Ivf();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
47.     #endif
48.     #if defined(MEDIAINFO_LXF_YES)
49.         delete Info; Info=new File_Lxf();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
50.     #endif
51.     #if defined(MEDIAINFO_MK_YES)
52.         delete Info; Info=new File_Mk();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
53.     #endif
54.     #if defined(MEDIAINFO_MPEG4_YES)
55.         delete Info; Info=new File_Mpeg4();         if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
56.     #endif
57.     #if defined(MEDIAINFO_MPEGPS_YES)
58.         delete Info; Info=new File_MpegPs();         if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
59.     #endif
60.     #if defined(MEDIAINFO_MPEGTS_YES)
61.         delete Info; Info=new File_MpegTs();         if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
62.     #endif
63.     #if defined(MEDIAINFO_MPLI_YES)
64.         delete Info; Info=new File_Mpli();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
65.     #endif
66.     #if defined(MEDIAINFO_MXF_YES)
67.         delete Info; Info=new File_Mxf();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
68.     #endif
69.     #if defined(MEDIAINFO_NUT_YES)
70.         delete Info; Info=new File_Nut();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
71.     #endif
72.     #if defined(MEDIAINFO_OGG_YES)
73.         delete Info; Info=new File_Ogg();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
74.     #endif
75.     #if defined(MEDIAINFO_P2_YES)
76.         delete Info; Info=new File_P2_Clip();         if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
77.     #endif
78.     #if defined(MEDIAINFO_PMP_YES)
79.         delete Info; Info=new File_Pmp();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
80.     #endif
81.     #if defined(MEDIAINFO_RIFF_YES)
82.         delete Info; Info=new File_Riff();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
83.     #endif
84.     #if defined(MEDIAINFO_RM_YES)
85.         delete Info; Info=new File_Rm();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
86.     #endif
87.     #if defined(MEDIAINFO_SEQUENCEINFO_YES)
88.         delete Info; Info=new File_SequenceInfo();   if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
89.     #endif

```

```

89.     #endif
90.     #if defined(MEDIAINFO_SKM_YES)
91.         delete Info; Info=new File_Skm();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
92.     #endif
93.     #if defined(MEDIAINFO_SWF_YES)
94.         delete Info; Info=new File_Swf();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
95.     #endif
96.     #if defined(MEDIAINFO_TSP_YES)
97.         delete Info; Info=new File_MpegTs(); ((File_MpegTs*)Info)->TSP_Size=16; if (((Reader_File*)Reader)->Format_Test_PerParser(th
is, File_Name)>0) return 1;
98.     #endif
99.     #if defined(MEDIAINFO_WM_YES)
100.        delete Info; Info=new File_Wm();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
101.    #endif
102.    #if defined(MEDIAINFO_XDCAM_YES)
103.        delete Info; Info=new File_Xdcam_Clip();        if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
104.    #endif
105.    #if defined(MEDIAINFO_DPG_YES)
106.        delete Info; Info=new File_Dpg();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
107.    #endif
108.
109.    // Video
110.    #if defined(MEDIAINFO_AVC_YES)
111.        delete Info; Info=new File_Avc();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
112.    #endif
113.    #if defined(MEDIAINFO_HEVC_YES)
114.        delete Info; Info=new File_Hevc();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
115.    #endif
116.    #if defined(MEDIAINFO_AVS_V_YES)
117.        delete Info; Info=new File_AvsV();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
118.    #endif
119.    #if defined(MEDIAINFO_DIRAC_YES)
120.        delete Info; Info=new File_Dirac();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
121.    #endif
122.    #if defined(MEDIAINFO_FLIC_YES)
123.        delete Info; Info=new File_Flic();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
124.    #endif
125.    #if defined(MEDIAINFO_H263_YES)
126.        delete Info; Info=new File_H263();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
127.    #endif
128.    #if defined(MEDIAINFO_MPEG4V_YES)
129.        delete Info; Info=new File_Mpeg4v();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
130.    #endif
131.    #if defined(MEDIAINFO_MPEGV_YES)
132.        delete Info; Info=new File_Mpegv();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
133.    #endif
134.    #if defined(MEDIAINFO_VC1_YES)
135.        delete Info; Info=new File_Vc1();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
136.    #endif
137.    #if defined(MEDIAINFO_VC3_YES)
138.        delete Info; Info=new File_Vc3();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
139.    #endif
140.    #if defined(MEDIAINFO_Y4M_YES)
141.        delete Info; Info=new File_Y4m();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
142.    #endif
143.
144.    // Audio
145.    #if defined(MEDIAINFO_AAC_YES)
146.        delete Info; Info=new File_Aac();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
1;
147.    #endif
148.    #if defined(MEDIAINFO_AC3_YES)
149.        delete Info; Info=new File_Ac3();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
150.    #endif
151.    #if defined(MEDIAINFO_SMPTST0337_YES)
152.        delete Info; Info=new File_SmpteSt0337();        if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
153.    #endif
154.    #if defined(MEDIAINFO_ALS_YES)
155.        delete Info; Info=new File_Als();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;
156.    #endif
157.    #if defined(MEDIAINFO_AMR_YES)
158.        delete Info; Info=new File_Amr();                if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) return
n 1;

```

```

159.     #endif
160.     #if defined(MEDIAINFO_AMV_YES)
161.         delete Info; Info=new File_Amv();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
162.     #endif
163.     #if defined(MEDIAINFO_APE_YES)
164.         delete Info; Info=new File_Ape();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
165.     #endif
166.     #if defined(MEDIAINFO_AU_YES)
167.         delete Info; Info=new File_Au();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
168.     #endif
169.     #if defined(MEDIAINFO_DTS_YES)
170.         delete Info; Info=new File_Dts();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
171.     #endif
172.     // Too many false-positives
173.     // #if defined(MEDIAINFO_DOLBYE_YES)
174.     //     delete Info; Info=new File_DolbyE();       if (((Reader_File*)Reader)-
>Format_Test_PerParser(this, File_Name)>0) return 1;
175.     // #endif
176.     #if defined(MEDIAINFO_FLAC_YES)
177.         delete Info; Info=new File_Flac();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
178.     #endif
179.     #if defined(MEDIAINFO_IT_YES)
180.         delete Info; Info=new File_ImpulseTracker(); if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
181.     #endif
182.     #if defined(MEDIAINFO_LA_YES)
183.         delete Info; Info=new File_La();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
184.     #endif
185.     #if defined(MEDIAINFO_MIDI_YES)
186.         delete Info; Info=new File_Midi();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
187.     #endif
188.     #if defined(MEDIAINFO_MOD_YES)
189.         delete Info; Info=new File_Module();         if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
190.     #endif
191.     #if defined(MEDIAINFO_MPC_YES)
192.         delete Info; Info=new File_Mpc();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
193.     #endif
194.     #if defined(MEDIAINFO_MPCSV8_YES)
195.         delete Info; Info=new File_MpcSv8();        if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
196.     #endif
197.     #if defined(MEDIAINFO_MPEGA_YES)
198.         delete Info; Info=new File_Mpega();         if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
199.     #endif
200.     #if defined(MEDIAINFO_RKAU_YES)
201.         delete Info; Info=new File_Rkau();          if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retu
rn 1;
202.     #endif
203.     #if defined(MEDIAINFO_TAK_YES)
204.         delete Info; Info=new File_Tak();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
205.     #endif
206.     #if defined(MEDIAINFO_S3M_YES)
207.         delete Info; Info=new File_ScreamTracker3(); if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
208.     #endif
209.     #if defined(MEDIAINFO_TTA_YES)
210.         delete Info; Info=new File_Tta();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
211.     #endif
212.     #if defined(MEDIAINFO_TWINVQ_YES)
213.         delete Info; Info=new File_TwinVQ();        if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
214.     #endif
215.     #if defined(MEDIAINFO_WVPK_YES)
216.         delete Info; Info=new File_Wvpk();          if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
217.     #endif
218.     #if defined(MEDIAINFO_XM_YES)
219.         delete Info; Info=new File_ExtendedModule(); if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
220.     #endif
221.
222.     // Text
223.     #if defined(MEDIAINFO_N19_YES)
224.         delete Info; Info=new File_N19();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
225.     #endif
226.     #if defined(MEDIAINFO_SCC_YES)
227.         delete Info; Info=new File_Scc();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;

```



```

228.         #endif
229.         #if defined(MEDIAINFO_SUBRIP_YES)
230.             delete Info; Info=new File_SubRip();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
231.         #endif
232.         #if defined(MEDIAINFO_TTML_YES)
233.             delete Info; Info=new File_Ttml();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
234.         #endif
235.         #if defined(MEDIAINFO_OTHERTEXT_YES)
236.             delete Info; Info=new File_OtherText();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
237.         #endif
238.
239.         // Image
240.         #if defined(MEDIAINFO_BMP_YES)
241.             delete Info; Info=new File_Bmp();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
242.         #endif
243.         #if defined(MEDIAINFO_DPX_YES)
244.             delete Info; Info=new File_Dpx();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
245.         #endif
246.         #if defined(MEDIAINFO_GIF_YES)
247.             delete Info; Info=new File_Gif();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
248.         #endif
249.         #if defined(MEDIAINFO_ICO_YES)
250.             delete Info; Info=new File_Ico();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
251.         #endif
252.         #if defined(MEDIAINFO_JPEG_YES)
253.             delete Info; Info=new File_Jpeg();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
254.         #endif
255.         #if defined(MEDIAINFO_PCX_YES)
256.             delete Info; Info=new File_Pcx();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
257.         #endif
258.         #if defined(MEDIAINFO_PNG_YES)
259.             delete Info; Info=new File_Png();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
260.         #endif
261.         #if defined(MEDIAINFO_PSD_YES)
262.             delete Info; Info=new File_Psd();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
263.         #endif
264.         #if defined(MEDIAINFO_TIFF_YES)
265.             delete Info; Info=new File_Tiff();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
266.         #endif
267.         #if defined(MEDIAINFO_TGA_YES)
268.             delete Info; Info=new File_Tga();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
269.         #endif
270.
271.         // Archive
272.         #if defined(MEDIAINFO_ACE_YES)
273.             delete Info; Info=new File_Ace();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
274.         #endif
275.         #if defined(MEDIAINFO_7Z_YES)
276.             delete Info; Info=new File_7z();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
277.         #endif
278.         #if defined(MEDIAINFO_BZIP2_YES)
279.             delete Info; Info=new File_Bzip2();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
280.         #endif
281.         #if defined(MEDIAINFO_ELF_YES)
282.             delete Info; Info=new File_Elf();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
283.         #endif
284.         #if defined(MEDIAINFO_GZIP_YES)
285.             delete Info; Info=new File_Gzip();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
286.         #endif
287.         #if defined(MEDIAINFO_ISO9660_YES)
288.             delete Info; Info=new File_Iso9660();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
289.         #endif
290.         #if defined(MEDIAINFO_MZ_YES)
291.             delete Info; Info=new File_Mz();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
292.         #endif
293.         #if defined(MEDIAINFO_RAR_YES)
294.             delete Info; Info=new File_Rar();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
295.         #endif
296.         #if defined(MEDIAINFO_TAR_YES)
297.             delete Info; Info=new File_Tar();             if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur

```

```

n 1;
298.     #endif
299.     #if defined(MEDIAINFO_ZIP_YES)
300.         delete Info; Info=new File_Zip();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
301.     #endif
302.
303.     // Other
304.     #if !defined(MEDIAINFO_OTHER_NO)
305.         delete Info; Info=new File_Other();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
306.     #endif
307.
308.     // Default (empty)
309.         delete Info; Info=new File_Unknown();           if (((Reader_File*)Reader)->Format_Test_PerParser(this, File_Name)>0) retur
n 1;
310.     return 0;
311. }
```

至此Open()函数的分析完毕。

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/12449803>

文章标签： [mediainfo](#) [open](#) [函数分析](#) [源代码](#)

个人分类： [MediaInfo](#)

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