**常用网页播放器代码**

我们在网页上看到的播放器无外乎WMP/RealOne/Macromedia Flash Player,其他的无非是面板不同，或者添加了其他控件，对于计算机上安装的一些播放器也都是编码和×××的整合，其最核心的编码和解码技术是相同的。例如：网络上最流行的windows media流(asf,wma,wmv格式...)，Real流(rm,rmvb...)，还有MPEG系列编码格式(MP4/MP3格式...)  
Windows Media Video 是微软推出的一种流媒体格式，它是在“同门”的ASF（Advanced Stream Format）格式升级延伸来得.在同等视频质量下，WMV格式的体积非常小，因此很适合在网上播放和传输。Windows Media Player9兼容所有格式的WMV,官方编码器是Windows Media Encoder ，但是如果你想转制 高质量的wmv文件，那您一定要有超大的内存来处理数据...  
无意中发现CASTPOST的播放器可以自己定义大小，对于WMV格式的在线播放可以说已经足够快了,然后就费了好大劲把一些精彩的短片和一些经典的MTV转化WMV格式放了上来,尽管现在不能下载了，但是只要不是连接人数过多，播放起来还是很流畅的^\_^  
  
WMP加入了ActiveX×××控件，不仅可以放曲子，还能放Flash和其它视频文件  
<object align=middle classid="CLSID:22d6f312-b0f6-11d0-94ab-0080c74c7e95" class=OBJECT id=MediaPlayer width=196 height=196>  
<param name=ShowStatusBar value=0>  
<param name=Filename value="[[url]http://202.116[/url]](http://202.0.0.116/).\*.\*/video/story/chinese/hynh/b.wmv">  
<embed type=application/x-oleobject codebase="[[url]http://activex.microsoft.com/activex/con[/url]](http://activex.microsoft.com/activex/con) ... n/nsmp2inf.cab#Version=5,1,52,701" >  
</embed>  
</object>  
想用WMP连续播放请参照ASX元文件使用讲解：使用ASX播放列表吧  
  
上面的这个播放器是老式的那种，6.4版本！新式播放器是在MediaPlayer9.0以后出现的，也就是说只有装了9.0或9.0以上的播放器才能正常使用的。  
  
  
下面是新式播放器代码，相对以前的来说要简单很多：  
<object id="player" height="64" width="260" classid="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6">  
<param NAME="AutoStart" VALUE="-1">  
<!--是否自动播放-->  
<param NAME="Balance" VALUE="0">  
<!--调整左右声道平衡,同上面旧播放器代码-->  
<param name="enabled" value="-1">  
<!--播放器是否可人为控制-->  
<param NAME="EnableContextMenu" VALUE="-1">  
<!--是否启用上下文菜单-->  
<param NAME="url" value="/blog/1.wma">  
<!--播放的文件地址-->  
<param NAME="PlayCount" VALUE="1">  
<!--播放次数控制,为整数-->  
<param name="rate" value="1">  
<!--播放速率控制,1为正常,允许小数,1.0-2.0-->  
<param name="currentPosition" value="0">  
<!--控件设置:当前位置-->  
<param name="currentMarker" value="0">  
<!--控件设置:当前标记-->  
<param name="defaultFrame" value="">  
<!--显示默认框架-->  
<param name="invokeURLs" value="0">  
<!--脚本命令设置:是否调用URL-->  
<param name="baseURL" value="">  
<!--脚本命令设置:被调用的URL-->  
<param name="stretchToFit" value="0">  
<!--是否按比例伸展-->  
<param name="volume" value="50">  
<!--默认声音大小0%-100%,50则为50%-->  
<param name="mute" value="0">  
<!--是否静音-->  
<param name="uiMode" value="mini">  
<!--播放器显示模式:Full显示全部;mini最简化;None不显示播放控制,只显示视频窗口;invisible全部不显示-->  
<param name="windowlessVideo" value="0">  
<!--如果是0可以允许全屏,否则只能在窗口中查看-->  
<param name="fullScreen" value="0">  
<!--开始播放是否自动全屏-->  
<param name="enableErrorDialogs" value="-1">  
<!--是否启用错误提示报告-->  
<param name="SAMIStyle" value>  
<!--SAMI样式-->  
<param name="SAMILang" value>  
<!--SAMI语言-->  
<param name="SAMIFilename" value>  
<!--字幕ID-->  
</object>  
  
  
--------------------------------------------------------------------------------  
  
  
RealOne播放器代码：  
<object id="vid" classid="clsid:CFCDAA03-8BE4-11cf-B84B-0020AFBBCCFA" width=427 height=300>  
<param name="\_ExtentX" value="11298">  
<param name="\_ExtentY" value="7938">  
<param name="AUTOSTART" value="-1">  
<param name="SHUFFLE" value="0">  
<param name="PREFETCH" value="0">  
<param name="NOLABELS" value="-1">  
<param name="SRC" value="[' target=\_blank>[url]rtsp://211.89.225.1/encoder/cnr3[/url]";>](rtsp://211.89.225.1/encoder/cnr3)  
<param name="CONTROLS" value="Imagewindow">  
<param name="CONSOLE" value="clip1">  
<param name="LOOP" value="0">  
<param name="NUMLOOP" value="0">  
<param name="CENTER" value="0">  
<param name="MAINTAINASPECT" value="0">  
<param name="BACKGROUNDCOLOR" value="#000000">  
</object> <object id="vid2" classid="clsid:CFCDAA03-8BE4-11cf-B84B-0020AFBBCCFA" width=427 height=30>  
<param name="\_ExtentX" value="11298">  
<param name="\_ExtentY" value="794">  
<param name="AUTOSTART" value="-1">  
<param name="SHUFFLE" value="0">  
<param name="PREFETCH" value="0">  
<param name="NOLABELS" value="-1">  
<param name="SRC" value="[' target=\_blank>[url]rtsp://211.89.225.1/encoder/cnr3[/url]";>](rtsp://211.89.225.1/encoder/cnr3)  
<param name="CONTROLS" value="ControlPanel">  
<param name="CONSOLE" value="clip1">  
<param name="LOOP" value="0">  
<param name="NUMLOOP" value="0">  
<param name="CENTER" value="0">  
<param name="MAINTAINASPECT" value="0">  
<param name="BACKGROUNDCOLOR" value="#000000">  
</object>  
  
--------------------------------------------------------------------------------  
  
看着很多很复杂的样子，其实就是多了些播放器参数，可以不要...  
Real Player 网页播放器 参数含义  
  
参数：autostart 属性：True或是False 作用：指定是否自动播放指定的源文件  
参数：backgroundcolor 属性：任何用符号“#”开头的16进制数值或是任何预定义的颜色作用：指定图像窗口的背景颜色  
参数：center 属性：True或是False 作用：指定片断使用初始编码大小播放，并且在图像窗口的中央。  
参数：classid 属性："clsid：CFCDAA03-8BE4-1lcf-B84B0020AFBBCCFA:\*\* 作用：用于指定ActiveX控件的唯一的字符串标示，可以认出嵌入的RealPalyer播放器。  
参数：console 属性：任何字符串作用：可以将各种不同的RealPlayer控制聚集在网页上，这样它们可以交互使用或是保持独立，而且互相不影响  
参数：controls 属性：ImageWindow，All，ControlPanel，PlavButton，PlayOnlyButton， PauseButton,StopButton，FFCtrl，RWCtrl，MuteCtrl，MuteVolume，VolumeSlider，PositionSlider,TACCtrl，HomeCtrl，InfoVolumePanel，InfoPanel，StatusBar，StatusField，PositionField 作用：可以让你指定那些控制是可见的。  
参数：height 属性：任何整数值作用：指定RealPlayer元素的高度，单位:像素  
参数：id 属性：任何字符串作用：为标签中的RealPlayer元素指定名字。  
参数：p\_w\_picpathstatus 属性：True或是False 作用：指定是否在图像窗口中显示状态信息，默认值是true  
参数：loop 属性：True或是False 作用：可以让你指定片断是否无限循环  
参数：maintainaspect 属性：True或是False 作用：默认RealPlayer拉伸所有的片断来充满整个图像窗口。  
参数：name 属性：任何字符串作用：为标签中的RealPlayer元素指定名字(在标签中使用id)  
参数：nojava 属性：True或是False 作用：避免启动Java虚拟机  
参数：nolabels 属性：True或是False 作用：可以禁止显示标题或是版权信息(realplayer5.0以上时，它是垃圾...)  
参数：nologo 属性：True或是False 作用：避免RealPlayer启动时在图像窗口中显示  
参数：numloop 属性：任何整数值作用：让你能够指定文件片循环的次数，不需要参数loop  
参数：prefetch 属性：True或是False 作用：指定在播放前，RealPlayer是否可以获得流描述信息，默认值是False  
参数：region 属性：任何字符串作用：同SMIL一起使用。允许你指定使用HTML代替SMIL  
参数：scriptcallbacks 属性：用逗号分割的列表作用：指定浏览器的回调监控（好高级的东东！）  
参数：shuffle 属性：True或是False 作用：同多文件片的ram文件或是SMIL文件一起使用。可以让RealPlayer随机播放列表中的文件  
参数：src 属性：任何合法的相对或是完整的URL 作用：指定播放的文件或是源文件的地址  
参数：type 属性：字符串作用：为嵌入插件指定MIME类型  
参数：width 属性：任何整数值作用：指定RealPlayer元素的宽度

|  |
| --- |
| RealPlayer的一些函数、方法和过程  这是 Real Player ActiveX Control Library (Version 1.0) 的所有函数与方法，有兴趣可以研究一下。  function GetSource: WideString; procedure SetSource(const lpszNewValue: WideString); function GetConsole: WideString; procedure SetConsole(const lpszNewValue: WideString); function GetControls: WideString; procedure SetControls(const lpszNewValue: WideString); function GetNoLabels: WordBool; procedure SetNoLabels(bNewValue: WordBool); function GetAutoStart: WordBool; procedure SetAutoStart(bNewValue: WordBool); function GetAutoGotoURL: WordBool; procedure SetAutoGotoURL(bNewValue: WordBool); function GetVolume: Smallint; procedure SetVolume(nVol: Smallint); function GetMute: WordBool; procedure SetMute(bMute: WordBool); function GetLoop: WordBool; procedure SetLoop(bVal: WordBool); function GetImageStatus: WordBool; procedure SetImageStatus(bEnable: WordBool); function GetPacketsTotal: Integer; function GetPacketsReceived: Integer; function GetPacketsOutOfOrder: Integer; function GetPacketsMissing: Integer; function GetPacketsEarly: Integer; function GetPacketsLate: Integer; function GetBandwidthAverage: Integer; function GetBandwidthCurrent: Integer; procedure DoPlayPause; procedure DoStop; procedure DoNextItem; procedure DoPrevItem; function CanPlayPause: WordBool; function CanStop: WordBool; function HasNextItem: WordBool; function HasPrevItem: WordBool; function HasNextEntry: WordBool; function HasPrevEntry: WordBool; procedure DoNextEntry; procedure DoPrevEntry; procedure AboutBox; procedure EditPreferences; procedure HideShowStatistics; function IsStatisticsVisible: WordBool; procedure DoGotoURL(const url: WideString; const target: WideString); procedure DoPlay; procedure DoPause; function GetPosition: Integer; function GetPlayState: Integer; function GetLength: Integer; function GetTitle: WideString; function GetAuthor: WideString; function GetCopyright: WideString; function GetClipWidth: Integer; function GetClipHeight: Integer; function CanPlay: WordBool; function CanPause: WordBool; procedure SetPosition(lPosition: Integer); function GetNumLoop: Integer; procedure SetNumLoop(lVal: Integer); function GetCenter: WordBool; procedure SetCenter(bVal: WordBool); function GetNoLogo: WordBool; procedure SetNoLogo(bVal: WordBool); function GetMaintainAspect: WordBool; procedure SetMaintainAspect(bVal: WordBool); function GetBackgroundColor: WideString; procedure SetBackgroundColor(const pVal: WideString); function GetStereoState: WordBool; function GetLiveState: WordBool; function GetShowStatistics: WordBool; procedure SetShowStatistics(bVal: WordBool); function GetShowPreferences: WordBool; procedure SetShowPreferences(bVal: WordBool); function GetShow WordBool; procedure SetShowAbout(bVal: WordBool); function GetOriginalSize: WordBool; procedure SetOriginalSize; function GetDoubleSize: WordBool; procedure SetDoubleSize; function GetFullScreen: WordBool; procedure SetFullScreen; function GetEnableContextMenu: WordBool; procedure SetEnableContextMenu(bVal: WordBool); function GetEnableOriginalSize: WordBool; procedure SetEnableOriginalSize(bVal: WordBool); function GetEnableDoubleSize: WordBool; procedure SetEnableDoubleSize(bVal: WordBool); function GetEnableFullScreen: WordBool; procedure SetEnableFullScreen(bVal: WordBool); function GetEnableMessageBox: WordBool; procedure SetEnableMessageBox(bVal: WordBool); procedure SetTitle(const pVal: WideString); procedure SetAuthor(const pVal: WideString); procedure SetCopyright(const pVal: WideString); function GetWantKeyboardEvents: WordBool; procedure SetWantKeyboardEvents(bWantsEvents: WordBool); function GetWantMouseEvents: WordBool; procedure SetWantMouseEvents(bWantsEvents: WordBool); function GetNumEntries: Smallint; function GetCurrentEntry: Smallint; function GetEntryTitle(uEntryIndex: Smallint): WideString; function GetEntryAuthor(uEntryIndex: Smallint): WideString; function GetEntryCopyright(uEntryIndex: Smallint): WideString; function GetEntryAbstract(uEntryIndex: Smallint): WideString; procedure SetCanSeek(bCanSeek: WordBool); function GetCanSeek: WordBool; function GetBufferingTimeElapsed: Integer; function GetBufferingTimeRemaining: Integer; function GetConnectionBandwidth: Integer; function GetPreferedLanguageString: WideString; function GetPreferedLanguageID: Integer; function GetUserCountryID: Integer; function GetNumSources: Smallint; function GetSourceTransport(nSourceNum: Smallint): WideString; function GetWantErrors: WordBool; procedure SetWantErrors(bVal: WordBool); function GetShuffle: WordBool; procedure SetShuffle(bVal: WordBool); function GetVersionInfo: WideString; function GetLastMessage: WideString; function GetLastErrorSeverity: Integer; function GetLastErrorRMACode: Integer; function GetLastErrorUserCode: Integer; function GetLastErrorUserString: WideString; function GetLastErrorMoreInfoURL: WideString; procedure SetPreFetch(bVal: WordBool); function GetPreFetch: WordBool; procedure SetRegion(const pVal: WideString); function GetRegion: WideString; function GetIsPlus: WordBool; function GetConsoleEvents: WordBool; procedure SetConsoleEvents(bVal: WordBool); function GetDRMInfo(const pVal: WideString): WideString; property ControlInterface: IRealAudio read GetControlInterface; property DefaultInterface: IRealAudio read GetControlInterface; |

网页中wmp视频控制之使用指南  
  
//wmplayer.controls.currentPosition = wmplayer.currentMedia.duration \* (i / 100)  
//  
//  
//  
□播放方法和属性  
除播放、暂停、停止外，媒体播放器还可用下面的属性：  
扫描(Scanning)——与录像机的快进快倒功能类似；  
搜索(Seeking)——直接移到剪辑中标记的特定上演时间；  
  
□播放  
媒体播放器提供了两种技术来指定要播放的媒体名称(剪辑)，你可以设置 FileName 属性，或调用 Open 方法。 如果 AutoStart 属性的值是 true 的话，当 FileName 属性被设置成该电影剪辑的 URL 后，该电影剪辑将开始播放；否则电影剪辑不会开始播放， 除非你调用 Play 方法。 Open 方法开始播放是异步的，它不象 Play 方法要等到其他的进程结束后才开始播放。  
媒体播放器提供了下面的类似录像机的属性和方法来控制流媒体的播放：  
Play、Stop、Pause 方法，来开始、停止、暂停流媒体。  
PlayCount 属性，设置文件播放的次数。  
AutoRewind 属性，确定当停止播放时是否返回到电影剪辑的开始部分。  
  
□音频控制  
媒体播放器提供了下面的属性来管理音频：  
Balance 属性，确定左右音箱的声音平衡；  
Volume 属性，用来加大或降低音量；  
Mute 属性，用来关闭或打开声音；  
※你可以将 ShowAudioControls 属性设为 true 来在控制栏添加处理声音的控件。  
  
□扫描  
媒体播放器提供了下面的属性用以扫描：  
FastForward 方法，快进；  
FastReverse 方法，快倒；  
Rate属性，改变播放速率；  
※要想使电影剪辑能被扫描，必须将 CanScan 和 AllowScan 属性设为 true。  
  
□搜索  
用于搜索的属性有：  
MarkerCount 属性，指剪辑中标记的总数量；  
CurrentMaker、GetMarkerName、GetMarkerTime 方法，用于返回标记信息；  
MarkerHit 事件，当遇到标记时触发；  
CurrentPosition 属性，当前位置(用秒度量)，可用来将播放头移到剪辑中指定的点；  
PositionChange 事件，当设置 CurrentPosition 属性时触发；  
※要想搜索到任意的时间，必须将 CanSeek 属性设为 true，要想搜索到标记点，必须将 CanSeekToMarkers 属性设为 true。  
  
□媒体播放器的外观界面  
在网页中，你可以通过相关属性来控制媒体播放器的哪些部分出现，哪些部分不出现。  
媒体播放器包括如下元素：  
Video Display Panel：视频显示面板；  
Video Border：视频边框；  
Closed Captioning Display Panel；字幕显示面板；  
Track Bar；搜索栏；  
Control Bar with Audio and Position Controls：带有声音和位置控制的控制栏；  
Go To Bar：转到栏；  
Display Panel：显示面板；  
Status Bar：状态栏；  
下面的属性用来决定显示哪一个元素：  
ShowControls 属性：是否显示控制栏（包括播放控件及可选的声音和位置控件）；  
ShowAudioControls 属性：是否在控制栏显示声音控件（静音按钮和音量滑块）；  
ShowPositionControls 属性：是否在控制栏显示位置控件（包括向后跳进、快退、快进、向前跳进、预览播放列表中的每个剪辑）；  
ShowTracker 属性：是否显示搜索栏；  
ShowDisplay 属性：是否显示显示面板（用来提供节目与剪辑的信息）；  
ShowCaptioning 属性：是否显示字幕显示面板；  
ShowGotoBar 属性：是否显示转到栏；  
ShowStatusBar 属性：是否显示状态栏；  
  
□播放列表  
媒体播放器提供下面的方法来访问播放列表中的剪辑：  
Next 方法，跳到节目（播放列表）中的下一个剪辑；  
Previous 方法，跳回到节目中的上一个剪辑；  
媒体播放器的一个特性是能够预览节目中的每一个剪辑，使用如下属性：  
PreviewMode 属性，决定媒体播放器当前是否处于预览模式；  
CanPreview 属性，决定媒体播放器能否处于预览模式；  
在windows 媒体元文件中，可以为每一个剪辑指定预览时间——PREVIEWDURATION,如果没有指定，那么默认的预览时间是10秒钟。  
你也可以用Windows 媒体元文件来添加 watermarks 与 banners，元文件也支持插入广告时的无间隙流切换。  
  
□节目信息  
使用 GetMediaInfoString 方法可以返回相关剪辑或节目的如下信息：  
文件名：File name  
标题：Title  
描述：Description  
作者：Author  
版权：Copyright  
级别：Rating  
URLs：logo icon、watermark、banner的地址  
剪辑信息可以放在媒体文件中，也可以放在Windows 媒体元文件中，或者两者都放。

剪辑信息可以放在媒体文件中，也可以放在windows 媒体元文件中，或者两者都放。如果在元文件中指定了剪辑信息，那么用 GetMediaInfoString 方法返回的就是元文件中的信息，而不会返回剪辑中包含的信息。  
在元文件中，附加信息可以放置在每一个剪辑或节目的 PARAM标签中。你可以为每个剪辑添加任意多个 PARAM 标签，用来存储自定义的信息或链接到相关站点。在 PARAM 标签中的信息可以通过 GetMediaParameter 方法来访问。  
下面的属性返回有关大小和时间的信息：  
ImageSourceHeight、ImageSourceWidth：返回图像窗口的显示尺寸；  
Duration 属性，返回剪辑的长度(秒)， 要检测这个属性是否包含有效的数值，请检查IsDurationValid 属性。(对于广播的视频，其长度是不可预知的)。  
  
□字幕  
你可以用 .smi 文件来为你的节目添加字幕。媒体播放器支持下面的属性来处理字幕：  
SAMIFileName 属性，指定 .smi 文件的名字；  
SAMILang 属性，指定字幕的语言(如果没有指定则使用第一种语言)；  
SAMIStyle 属性，指定字幕的文字大小和样式；  
ShowCaptioning 属性，决定是否显示字幕显示面板；  
  
□脚本命令  
伴随音频、视频流，你可以在流媒体文件中加入脚本命令。脚本命令是多媒体流中与特定时间同步的多对Unicode串。第一个串标识待发命令的类型，第二个串指定要执行的命令。  
当流播放到与脚本相关的时间时，控件会向网页发送一个 scriptCommand事件，然后由事件处理进程来响应这个事件。脚本命令字符串会作为脚本命令事件的参数传递给事件处理器。  
媒体播放器会自动处理下面类型的内嵌脚本命令：  
1)URL型命令：当媒体播放器控件收到一个URL型的命令时，指定的URL会被装载到用户的默认浏览器。如果媒体播放器嵌在一个分帧的HTML文件中，URL页可以装载到由脚本命令指定的帧内。如果脚本命令没有指定一个帧，将由 DefaultFrame 属性决定将 URL 页装入哪一帧。  
你可以通过设置 InvokeURLs 属性来决定是否自动处理 URL 型的脚本命令。如果这个属性的值为 false ，媒体播放器控件将忽视 URL型命令。但是脚本命令事件仍会触发，这就允许你有选择地处理 URL 型命令。  
URL 型命令指定的是 URL 的相对地址。基地址是由 BaseURL属性指定的。媒体播放器控件传送的脚本命令事件的命令参数是链接好的地址。  
2)FILENAME型命令：当媒体播放器控件收到一个FILENAME型的命令时，它将 FileName属性设置为脚本命令提供的文件，之后媒体播放器会打开这个文件开始播放。 媒体播放器控件总是自动处理 FILENAME 型命令，不象 URL 型命令，它们不能被禁止。  
3)TEXT型命令：当媒体播放器控件收到一个 TEXT型的命令时，它会将命令的内容显示在控件的字幕窗口。内容可以是纯文本的，也可以是 HTML。  
4)EVENT型命令：当媒体播放器控件收到一个 EVENT型的命令时，它会在媒体元文件中搜索 EVENT 元素的 NAME 属性。如果 NAME 属性与脚本命令中的第二个字符串匹配，媒体播放器控件就执行包含在 EVENT 元素中的条目。  
5)OPENEVENT型命令：当媒体播放器控件收到一个 OPENEVENT型的命令时，它会在媒体元文件中检查 EVENT 元素，并打开匹配的标题，但不播放，直到收到来自 EVENT型命令的同名真实事件。  
  
□捕捉键盘和鼠标事件  
EnableContextMenu 与 ClickToPlay 属性为用户提供了在图像窗口进行操作的方法。  
如果 EnableContextMenu 属性为 true ，在图像窗口右击鼠标可以打开关联菜单，如果将ClickToPlay 属性设为 true ，用户可以单击图像窗口进行播放与暂停的切换。  
要接收鼠标移动和单击事件，请将 SendMouseMoveEvents 和 SendMouseClickEvents 属性设为 true 。鼠标事件有：  
MouseDown，当用户按下鼠标时产生；  
MouseUp，当用户释放鼠标时产生；  
MouseMove，当用户移动鼠标时产生；  
Click，当用户在媒体播放器上单击鼠标按钮时产生；  
DbClick，当用户在媒体播放器上双击鼠标按钮时产生；  
要接收键盘事件，请将 SendKeyboardEvents 属性设为 true 。键盘事件有：  
KeyDown，当用户按下一个键时产生；  
KeyUp，当用户释放一个键时产生；  
KeyPress，当用户按下并释放一个键时产生；  
  
□监测流状态与网络链接  
流状态属性包括：  
PlayState：播放状态；  
OpenState：打开状态；  
Bandwidth：带宽；

监测流状态与网络链接  
流状态属性包括：  
PlayState：播放状态；  
OpenState：打开状态；  
Bandwidth：带宽；  
支持的事件有：  
OpenStateChange：打开状态改变(仅当SendOpenStateChangeEvents属性为true时触发)  
PlayStateChange：播放状态改变(仅当SendPlayStateChangeEvents属性为true时触发)  
EndOfStream：流结束时触发；  
NewStream：打开新流时触发；  
网络接收属性包括：  
ReceptionQuality：接收质量；  
ReceivedPackets：已经收到的包；  
LostPackets：丢失的包；  
监测缓冲的属性有：  
BufferingTime：缓冲时间；  
BufferingCount：缓冲次数；  
BufferingProgress：缓冲进程；  
Buffering：缓冲事件；  
  
□错误处理  
媒体播放器提供了内建的错误处理功能——在对话框或状态栏显示错误信息。 另外，你可以自己添加错误处理程序。如果 SendErrorEvents 属性设置为 true，将不会显示错误框，而是发送错误事件；如果 SendErrorEvents 属性设置为 false，将显示错误框，而是发送错误事件。  
媒体播放器支持下面的错误处理事件：  
Error 事件，指有危险性错误发生；  
Warning 事件，指发生了非危险性的错误；  
当你的应用程序接收到一个错误事件，你可以检测下面的属性来确定具体的错误信息：  
HasError：检测目前的媒体播放器是否有错误；  
ErrorCode：提供与该类型错误相关的代码值；  
ErrorDescription：提供错误的描述信息；  
ErrorCorrection：指定媒体播放器对该类型的错误进行校正；  
  
□播放CD  
媒体播放器将 CD 看作（在每个音轨的开头具有标记的）单一音频流。 要在网页中使用CD，你需要将 FileName 属性设为 CDAUDIO:，必须带有冒号，如下所示：  
<HTML>  
<HEAD><TITLE>CD Audio Playback Example</TITLE></HEAD>  
<BODY>  
<OBJECT ID="MediaPlayer"  
CLASSID="CLSID:22d6f312-b0f6-11d0-94ab-0080c74c7e95"  
WIDTH="320"  
HEIGHT="508"  
STYLE="position:absolute; left:0px; top:70px;" >  
<PARAM NAME="FileName" VALUE="cdaudio:">  
<PARAM NAME="AutoStart" VALUE="0">  
<PARAM NAME="ShowControls" VALUE="1">  
<PARAM NAME="ShowStatusBar" VALUE="1">  
<PARAM NAME="ShowDisplay" VALUE="1">  
</OBJECT>  
</BODY>  
</HTML>

**相关文章**

**[Android实现媒体播放器](https://blog.51cto.com/yangshaoping/11420507" \t "_blank)**

[Android 提供了许多方法来控制音频/视频文件和流的播放。 其中一种方法是通过一个名为 MediaPlayer 的类。Android 正在提供 MediaPlayer 类来访问内置的媒体播放器服务，例如播放音频、视频等。 为了使用 MediaPlayer，我们必须调用这个类的静态方法 create()。 此方法返回 MediaPlayer 类的实例。 它的语法如下](https://blog.51cto.com/yangshaoping/11420507" \t "_blank)

androididexml

**[RTMP播放器的几种选择](https://blog.51cto.com/daniusdk/12032069" \t "_blank)**

[如何选择RTMP播放器？在选择RTMP播放器时，需要综合考虑多个因素，以确保选择的播放器能够满足实际需求并提供良好的用户体验。以下是一些选择RTMP播放器的建议：1. 功能需求低延迟：对于直播场景，低延迟是非常重要的。选择支持低延迟播放的RTMP播放器，以确保观众能够实时看到主播推送端的内容。多平台支持：考虑播放器是否支持跨平台播放，包括PC、移动设备等。这有助于扩大观众范围并提高播放的灵活性。音](https://blog.51cto.com/daniusdk/12032069" \t "_blank)

RTMP播放器RTMP playerwindows rtmp播放器linux rtmp播放器大牛直播SDK

**[常用网页播放器代码(3)](https://blog.51cto.com/u_15907753/5926262" \t "_blank)**

[RealPlayer网页播放面板（rm.rmvb.ram）音频播放器（无信息显示）音频播](https://blog.51cto.com/u_15907753/5926262" \t "_blank)

objectvideo链接地址数据音频播放

**[常用网页播放器代码(2)](https://blog.51cto.com/u_15907753/5971325" \t "_blank)**

[windows Media Player 9.0网页播放器](https://blog.51cto.com/u_15907753/5971325" \t "_blank)

filterwindowsobjectvideourl

**[常用网页播放器代码(1)](https://blog.51cto.com/u_15907753/5971323" \t "_blank)**

[WMP加入了ActiveX解码器控件，不仅可以放曲子，还能放Flash和其它视频文件](https://blog.51cto.com/u_15907753/5971323" \t "_blank)

functioninteger脚本objecturl

**[网页播放器](https://blog.51cto.com/yukai/363686" \t "_blank)**

[用到的两个网页播放器,embed那个兼容ie7，ie8，ff,object那个似乎只支持ie8，ie7 <embed id="MediaPlayer2" style="border-color:#0099FF; border-bottom-width:1px; border-left-width:1px;](https://blog.51cto.com/yukai/363686" \t "_blank)

职场休闲object网页播放器embed

**[网页嵌入播放器代码集锦](https://blog.51cto.com/u_2776699/5845333" \t "_blank)**

[<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"](https://blog.51cto.com/u_2776699/5845333" \t "_blank)

程序代码idemicrosoft

**[.net 网页播放器](https://blog.51cto.com/iammoon24/771231" \t "_blank)**

[.net 网页播放浅谈](https://blog.51cto.com/iammoon24/771231" \t "_blank)

.net网页播放器资料休闲

**[网页内嵌播放器](https://blog.51cto.com/962410314/1414103" \t "_blank)**

[<object id="bofang" classid="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6" width="280" height="265" codebase="http://activex.microsof](https://blog.51cto.com/962410314/1414103" \t "_blank)

网页播放器

**[Winamp网页播放器](https://blog.51cto.com/u_627724/6046901" \t "_blank)**

[哈哈，早上5点醒来睡不着，到www.real2000.org听歌，看到这个播放器不错，于是把它给扣了出来。^\_^    原先是分了几个js文件调用的，为了贴代码方便，偶都放在一个文件里了：）共有2个文件和几十个图片。    在线演示：http://music.lzr.com.cn/Apple/Winamp    打包下载：http://music.lzr.com.cn/Apple/File](https://blog.51cto.com/u_627724/6046901" \t "_blank)

functionborderimagetablejavascript

**[网页视频播放器](https://blog.51cto.com/myreed/385378" \t "_blank)**

[Flowplayer：http://blog.csdn.net/mygoare/archive/2010/06/21/5683563.aspxJW Player：http://www.longtailvideo.com/support/jw-player-setup-wizard?example=201](https://blog.51cto.com/myreed/385378" \t "_blank)

职场休闲FlowplayerJW Player

**[python网页播放器](https://blog.51cto.com/u_16213444/10538733" \t "_blank)**

[# Python网页播放器Python 是一种功能强大的编程语言，广泛应用于各种领域，包括Web开发。 在Web开发中，经常需要添加音频和视频播放功能，而Python提供了许多库和工具来实现这一目的。 在本文中，我们将介绍如何使用Python创建一个简单的网页播放器，并演示如何用代码实现。## 搭建环境在开始之前，我们需要安装一个用于Web开发的Python库，如Flask。Flask](https://blog.51cto.com/u_16213444/10538733" \t "_blank)

Python应用程序Web

**[网页播放器代码](https://blog.51cto.com/u_15127699/4177689" \t "_blank)**

[1.avi格式代码片断如下：<object id="video" width="400" height="200" border="0" classid="clsid:CFCDAA03-8BE4-11cf-B84B-0020AFBBCCFA"><param name="ShowDisplay" value="0"><param name="ShowControls" ...](https://blog.51cto.com/u_15127699/4177689" \t "_blank)

ide控件元文件字符串2d

**[网页播放器代码大全](https://blog.51cto.com/u_15127598/4242116" \t "_blank)**

[1.avi格式 代码片断如下： 2.mpg格式 代码片断如下： 3.smi格式 代码片断如下： 4.rm格式 代码片断如下： 5.wmv格式 代码片断如下： 6.wma格式 放在 里面。下面是部分解释: 7.Windows Media Player 系列(不同面板样式) 综合型： 简易型： 标签型: 多功能型： 不仅可以放曲子，还能放Flash和其它视频文件。 网页播放器的参...](https://blog.51cto.com/u_15127598/4242116" \t "_blank)

控件ide元文件2dmicrosoft

**[jquery网页音频播放器代码 html音频播放器](https://blog.51cto.com/u_12959/6384708" \t "_blank)**

[html音频音乐播放器插件效果制作](https://blog.51cto.com/u_12959/6384708" \t "_blank)

jquery网页音频播放器代码htmljQueryChrome

**[java servlet网页视频播放器代码 web视频播放器代码](https://blog.51cto.com/u_16099350/10887232" \t "_blank)**

[收集最常用的网页中嵌入视频代码大全,各种各样的网页视频播放器代码.网页中嵌入视频代码综合完全版　　1.avi格式　　代码片断如下：　　程序代码<objectid="video"width="400"height="200"border="0"classid="clsid:CFCDAA03-8BE4-11cf-B84B-0020AFBBCCFA">　　<paramname="Sh](https://blog.51cto.com/u_16099350/10887232" \t "_blank)

ide代码片程序代码

**[经常使用网页播放器代码](https://blog.51cto.com/u_15127544/4018040" \t "_blank)**

[经常使用网页播放器代码 我们在网页上看到的播放器无外乎WMP/RealOne/Macromedia Flash Player,其它的无非是面板不同，或者加入了其它控件，对于计算机上安装的一些播放器也都是编码和解码器的整合，其最核心的编码和解码技术是同样的。比如：网络上最流行的windows medi...](https://blog.51cto.com/u_15127544/4018040" \t "_blank)

ide控件链接地址元文件字符串

**[网页视频播放器 jquery 网页视频播放器推荐](https://blog.51cto.com/u_16099234/11008183" \t "_blank)**

[西瓜播放器是字节跳动推出的一款播放器。字节跳动的视频业务大多数是短视频，早期的时候我们在 video.js 基础上做二次开发。后来发现很多功能达不到我们的要求，比如自定义UI的成本、视频的清晰度无缝切换、视频流量的节省。考虑到当前点播依旧是mp4居多，我们做了个大胆的假设：在播放器端加载视频、解析视频、转换格式，让不支持分段播放的mp4动态支持，这样就无须转换源视频的格式，服务器端也无其他开销。在](https://blog.51cto.com/u_16099234/11008183" \t "_blank)

网页视频播放器 jqueryide初始化chrome