

MPEG-4 In A Web Page

It's easier than you may think

EMBEDDING video in a web page has many advantages over stand-alone desktop players. Beyond the ability to customize the look-and-feel of the viewer's experience, embedded players can be automatically installed, can provide viewing statistics via conventional web logs, and allow a certain degree of security.

Player Basics

Virtually all modern streaming video players for the Windows environment are built as "ActiveX Controls" (a Microsoft name for "Comm Objects") that fit into the DirectShow environment. As a result, when you install a stand-alone player you have usually installed the components necessary for embedding the video in a web page. But when you install an embedded player, you do not have the stand-alone player. A stand-alone player is typically a set of user interfaces ("skins") that use various additional components, including the player ActiveX.

There is little functional difference between an "embedded player" and a "stand alone player". Rather, there is a "player object" that is useful by itself, and an application that bundles this object with other features and functions. Player vendors, including Real, Apple, and Microsoft, all create a stand-alone player that implement additional features beyond what is reasonable for just the video player object. For example, the QuickTime player ActiveX and the Real player ActiveX can play video but they do not report statistics. The stand-alone player can report statistics because they include functions beyond the player object itself. This keeps the player object smaller, more "lightweight", and good appropriate for automatic download and installation.

You probably realize that for embedding video in a web page, we are interested in just the player object, conveniently packaged as an ActiveX control. But with many possible players, how does your computer identify which one to use?

GUID

Every ActiveX object has a Globally Unique Identification number, or GUID. For example, the GUID for QuickTime is “02BF25D5-8C17-4B23-BC80-D3488ABDDC6B”. When your browser sees this identifier, it knows which software (the player itself) to use.

Each player object has various parameters, properties and methods available that are unique to each player.

Plug-Ins

The VBrick StreamPlayer is unique. It is not a “player” at all! Rather, it is a group of special components that enable Windows Media Player to display live MPEG-1, MPEG-2 and MPEG-4. The group includes the VBrick Program Guide, Source Filters, and both MPEG-2 and MPEG-4 audio/video decoders.

Just like Windows Media has its very own GUID (6BF52A52-394A-11D3-B153-00C04F79FAA6), so do the VBrick components.

As long as the components are available on your computer, they can be called from a web page. Importantly, it does not matter the source of the web page...it could be a page you encounter on any public web site, or an html page saved on your local computer. Basically, the html code tells the browser to find a software component already on the computer, and then tells that component what to do. In this way, the browser doesn't really display video – it simply instructs a compatible component to do it.

But it is inconvenient to install components manually, especially if we are talking about serving the half-billion people connected to the Internet! Happily, ActiveX components can be automatically installed from any web server via a “CAB” file. You can think of a CAB file as a specialized “zip” file that includes installation instructions. When your browser encounters an object (a GUID) that includes a CODEBASE statement, your browser will do several things. First, it will check to see if that GUID is already present on your computer. Next, it will check the revision. If you don't already have the component, or if the component specified in the web page is a later version than what you have, your browser will automatically download and install it. Now, if there is a new version of the component, you need only change the revision date on

one web page, and all viewers will get the update the next time they visit your page!

The Code

Let's start with Windows Media Player. Here is the code for Media Player version 9:

```
<object classid="clsid:6BF52A52-394A-11D3-B153-00C04F79FAA6" id="wmplayer"
width="352" height="240">
  <param name="URL" value>
</object>
```

You can learn about each parameter from the Microsoft player SDK, but note that the first parameter is called "URL". If you want to display a live MPEG-2 video stream, you cannot simply enter the IP address of the stream because the player does not know how to play standard streams.

In order to play live MPEG-1, MPEG-2, or MPEG-4, you need the VBrick components. The components will "intercept" the URL, configure the necessary MPEG decoders, and tells the Media Player window to display the video.

The easiest thing to do is to include the VBrick components on the same page as Media Player. To install these components, the following code is included on the page:

```
<object classid="clsid:0249ED44-B640-45BD-8066-17F81BFDC050"
CODEBASE="http://www.vbrick.com/CAB/STREAMPLAYER1.cab#Version=1,0,0,2"
</object>
<object classid="clsid:85887165-031A-4297-BC4E-6B246C120B9C"
CODEBASE="http://www.vbrick.com/CAB/STREAMPLAYER4.cab#Version=1,0,0,2"
</object>
```

Note that the object includes a CODEBASE statement that will load the component directly from the VBrick web site.

It is not necessary to include this code on every page. In fact, it is perfectly acceptable to include the above objects only on one page, such as an entry portal, as long as the page is viewed at least once. Its only purpose is to install the components. Once the components are installed they are ready to work. However, it is good practice to include the code on every viewing page to ensure the viewer has the latest code, regardless of how they entered your site (remember, the

components are automatically downloaded and installed only if needed, and if they are already present and up to date, nothing happens).

As discussed earlier, if a desktop installation of VBrick StreamPlayer Plus is conducted, the components are already present, but including the above code will do no harm.

MPEG URL's

To display a live MPEG-1 or MPEG-2 video stream in Media Player, simply use the following for the URL:

```
vbricksys://ip=ipaddress&port=portnumber&license
```

If you have a multicast stream at 239.1.1.1 on port 4444, your code will look like this:

```
<html>
<head>
<Title> VBrick Sample </title>
<body>
<object classid="clsid:6BF52A52-394A-11D3-B153-00C04F79FAA6" id="wmplayer"
width="245" height="240">
<param name="URL" value="vbricksys://ip=239.1.1.1&port=4444&License=your license
string">
</object>
<object classid="clsid:0249ED44-B640-45BD-8066-17F81BFDC050"
CODEBASE="http://www.vbrick.com/CAB/STREAMPLAYER1.cab#Version=1,0,0,2"
</object>
</body>
</html>
```

When you view the page, live MPEG video is instantly displayed! An alert is issued if you do not use a valid VBrick license.

Displaying of MPEG-4 via RTSP unicast, is equally easy. For example, if your VBXcast has an IP address of 192.168.1.1, and your stream name is "vbrickvideo1" your html code would be:

```
<html>
<head>
<Title> VBrick Sample </title>
<body>

<object classid="clsid:6BF52A52-394A-11D3-B153-00C04F79FAA6" id="wmplayer"
width="245" height="240">
<param name="URL" value="vbrtsp://192.168.1.1/vbrickvideo1">
</object>
```

```
<object classid="clsid:85887165-031A-4297-BC4E-6B246C120B9C"  
  CODEBASE="http://www.vbrick.com/CAB/STREAMPLAYER4.cab#Version=1,0,0,2"  
</object>  
  
</body>  
</html>
```

Note that for MPEG-4, you simply use “vbrtsp://” (Microsoft does not appear to properly support the standard protocol). The special VBrick component will intercept the “vbrtsp://” and use the RTSP standard for communications with the video source.

Support for the reception of MPEG-4 multicast is more complex. You have three choices:

1. Configure your VBXcast to multicast your MPEG-4 video, then press the “Create SDP File” in your VBXcast and save the SDP file. Locate the SDP file on a web server. Use the full path name to this SDP file in the player’s URL parameter (e.g. in place of “vbrtsp://192.168.1.1/vbrickvideo1” above).
2. Use Javascript or VBscript to program the VBrick components to identify the multicast stream and launch the player. Details on scripting VBrick’s StreamPlayer are in the “VBrick Video Toolkit”.
3. Use StreamPlayer or StreamPlayer Plus that automatically displays a Program Guide of multicast video streams.

Internet Viewing

VBrick StreamPlayer components are ideal for use in any network where UDP streaming is allowed. However, because StreamPlayer does not support HTTP tunneling, it may not be the best choice for a universal “consumer” MPEG-4 video viewing. QuickTime may be used in to support both UDP and HTTP, although the viewer will need to manually set the “Streaming Transport” preference.

That HTTP streaming is of interest may seem odd when you realize that the VBXcast itself only delivers the more-efficient UDP¹. But when a VBXcast feeds a reflector server, both UDP and HTTP streaming is supported.

¹ Due to overhead, HTTP streaming uses 30% more bandwidth per stream than does UDP.

Embedded QuickTime

QuickTime version 6 supports MPEG-4 directly, and like many players may be automatically installed.

To embed QuickTime in a web page, use this code:

```
<OBJECT CLASSID="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"  
WIDTH="276" HEIGHT="189"  
CODEBASE="http://www.apple.com/qtactivex/qtplugin.cab">  
<PARAM name="SRC" VALUE="myvideo.mov">  
<PARAM name="AUTOPLAY" VALUE="true">  
<PARAM name="CONTROLLER" VALUE="false">  
<EMBED SRC="myvideo.mov" WIDTH="360" HEIGHT="250" AUTOPLAY="true"  
CONTROLLER="false"  
PLUGINSPAGE="http://www.apple.com/quicktime/download/">  
</EMBED>  
</OBJECT>
```

Information about each QuickTime parameter can be found on the Apple web site.

Note that embedded QuickTime wants to see the SRC parameter set to a “.mov” file, and not a “rtsp://” stream. In this example, the “.mov” file is simply a QuickTime “reference movie” that need contain nothing more than the “rtsp://” address of your live stream. You simply create this reference movie and place it on your web server and point the player to it as shown.

For multicast, simply create a SDP file from your VBXcast, place it on your web server, and point QuickTime to this file in place of the .mov file.

Audience Participation

Having live MPEG-4 video that you can broadcast to virtually everyone, anywhere is powerful. You can give a live presentation, tutorial, product launch, religious service, and much more.

But as you give a live talk, how can your audience ask questions?

There are at least four answers:

1. Use a telephone – have your audience call you, ask their question, then hang up. Like a “Radio Talk Show”, it is very necessary that the caller turn off their sound to avoid echo. The delay through the Internet streaming system can vary from a few, to many seconds.

Avoid using a speakerphone. Use of a telephone tends to defeat the purpose and the spirit of live webcasting, and should be used only when there is a very small audience or when there are no alternatives.

2. Use email – have your audience send questions to you via email. This is a good solution because it is universally supported.
3. Use Instant Messaging – form a “IM Community” using popular Instant Messaging services, free or otherwise. The disadvantage is this does not scale well to very large audiences, and not everyone may have access to the IM service (AOL, MS, Netscape, etc.).
4. Create Chat Page – There are many free chat room services that can be easily integrated along with your live MPEG video. Typically, this is implemented in a frame set where your video is one frame page and your chat page is another. You can easily host your own chat page using applications such as ASPChatWorks, available for free download from <http://www.freevbcode.com/ShowCode.Asp?ID=858>. VBrick does not endorse or support this application, but it is implemented on the VBrick web site and successfully used for live webcasting sessions. For a modest fee, superior Java Applets exist on the open market that provide moderated chat and other advanced features. A modestly competent webmaster can create a video/chat frame set in only a few hours.

Synchronized Graphics

As we’ve seen, with VBrick it is very easy to send great video with great sound to everyone in an enterprise, or via the Internet, everyone in the world. But many presentations benefit from the use of charts and graphics, such as PowerPoint slides. How can these be included in a live electronic presentation?

One way to do this is with VBrick’s free SCAT (Screen Capture And Transfer) tool and your web server. SCAT will automatically capture your desktop image (for example, a full screen PowerPoint show slide) and send the image to your web server when you press a key. The

name of the image is the same each time you do a capture, so the JPEG image on your server is always the one you last sent.

All you need to do is simply construct a web page that includes that image from your web server, and use a Java Applet that refreshes the image on a periodic basis. Along side that image, simply include your embedded video and you instantly have live video along with images that display in near real time under control of the presenter.

For networks using multicast, the same approach can be used, but inexpensive 3rd party software (such as www.videoalive.com) can deliver graphics via multicasting, saving web server cost and providing virtually infinite scale.

Many other 3rd party tools exist that can be used to deliver images along with your video.

Novice Instruction

You do not need webmaster skills, or even web authoring software, to embed video in a web page. Just follow these simple instructions:

1. Open your "My Documents" folder.
2. Select File -> New -> Text Document, or just open Notepad if you know how.
3. Copy the following into your new Notepad document:

```
<html>
<head>
<Title> VBrick Sample </title>
<body>
<object classid="clsid:6BF52A52-394A-11D3-B153-00C04F79FAA6" id="wmplayer"
width="245" height="240">
  <param name="URL" value="vbrtsp://192.168.1.1/vbrickvideo1">
</object>
<object classid="clsid:85887165-031A-4297-BC4E-6B246C120B9C"
CODEBASE="http://www.vbrick.com/CAB/STREAMPLAYER4.cab#Version=1,0,0,2"
</object>
<object classid="clsid:0249ED44-B640-45BD-8066-17F81BFDC050"
CODEBASE="http://www.vbrick.com/CAB/STREAMPLAYER1.cab#Version=1,0,0,2"
</object>
</body>
</html>
```


4. If you want to view MPEG-4 from your VBXcast, edit the line that says value="vbrtsp://192.168.1.1/vbrickvideo1" and change it to your VBXcast IP address and stream name. If you want to view live MPEG-1 or MPEG-2 video sent via multicast, edit the line to read "vbricksys://ip=ipaddress&port=port".
5. Select "Save As" and save the document with any name you like.
6. Rename the document with a ".htm" extension.
7. Double-click on the document. Your IE browser will launch and you will see your live video.

Security

While not foolproof, your web page with embedded video can easily be made more secure by simply password-protecting it. By locating your web page in a password-protected directory on your web server, or through the use of ASP, CGI or similar, your page may only be accessible to those with the password. Of course, anyone who knows the address of the stream itself can still gain access, but it would be very difficult to guess this address. For example, if your RTSP stream name was "/xyz123" instead of "/sales", it would be very difficult to guess. Thus, the name of the stream becomes another "password", and if you have reasonable assurance that only those whom you have given access can view your video.

Making multicast stream secure is more problematic, but once again a reasonable degree of security is available by simply not broadcasting the Program Guide, using obscure addresses and port numbers, protecting the MPEG-4 SDP file, and/or using the VBrick Category feature.

Quick Reference

Item	GUID	Comments
StreamPlayer Package 1	0249ED44-B640-45BD-8066-17F81BFDC050	Package of multiple components needed to decode MPEG-1 and MPEG-2, and to provide multicast Program Guide.
StreamPlayer Package 4	85887165-031A-4297-BC4E-6B246C120B9C	Package of multiple components needed to decode MPEG-4.
VBexchange component	BB370BA2-725A-47e3-9459-0EF542B2FFC4	Component to detect and format MPEG-4 multicast with script.
VBrick Program Guide	0005C928-1B6D-43fb-BAE0-3783E03FA9CF	Component to detect and display live MPEG-1, MPEG-2 and MPEG-4 Program Guide
Windows Media Player 9	6BF52A52-394A-11D3-B153-00C04F79FAA6	Video player from Microsoft
QuickTime 6	02BF25D5-8C17-4B23-BC80-D3488ABDDC6B	MPEG-4 player from Apple
RealOne	CFCDA03-8BE4-11CF-B84B-0020AFBBCCFA	Requires 3 rd party plug-in to display MPEG-4