

AcroTeX.Net

## The rmannot Package Using VideoPlayerPlus

D. P. Story

A source file containing the example presented in this article is attached. Open the [attachments panel](#) (by clicking on the red link) and save to your local computer. There is a link on the [AeB Blog](#) page to download the FLV files used in this article.

## Table of Contents

<b>1 Introduction</b>	<b>3</b>
<b>2 Using VideoPlayerPlus.swf</b>	<b>3</b>
2.1 The playPlusX() Function . . . . .	6
2.2 Other Variations . . . . .	7

## 1. Introduction

When you play a FLV file, the SWF file `VideoPlayer.swf` is embedded in the PDF. It is `VideoPlayer.swf` that plays the FLV file. It is this SWF file that allows us to customize the look of the RMA, what skin to use, skin color, skin opacity, value, speed, and so on.

The `VideoPlayer.swf` file, which is shipped with Acrobat Pro, version 9 or later, lacks several useful features, among these are the ability to play more than one video in the same rich media annotation (RMA).

In the past year, there have been two extensions to Adobe's `VideoPlayer.swf`:

- `VideoPlayerPlus.swf` is available from Joel Geraci's web site [The PDF Developer Junkie Blog](#). Joel is a guru at Adobe. Extended features are in the form of additional JavaScript API to play more than one video in an RMA, change skins, change skin color, and a few others. Full documentation can be found on the reference blog page.

I have extensively tested the `VideoPlayerPlus.swf`, and it seems to work as advertised, no problem.

- `VideoPlayerX.swf` is another extension to the video player shipped by Adobe. This one is being developed by [UVSAR](#). Full documentation can be found on this page. This player offers a different set of JavaScript APIs, with some overlap with Joel's **PlayerPlus** widget.

The two referenced pages give instructions on how to use these video players with the user interface (UI).

I have extensively tested both widgets and have incorporated their use into `rmannot` system; instructions for installing `VideoPlayerPlus.swf` and `VideoPlayerX.swf` are covered in the [rmannot manual](#).

In this article, we discuss the use of `VideoPlayerPlus.swf`, in the next we look at `VideoPlayerX.swf`.

## 2. Using `VideoPlayerPlus.swf`

To use `VideoPlayerPlus.swf` to play FLV files instead of `VideoPlayer.swf`, place the command

```
\useVideoPlayerPlus
```

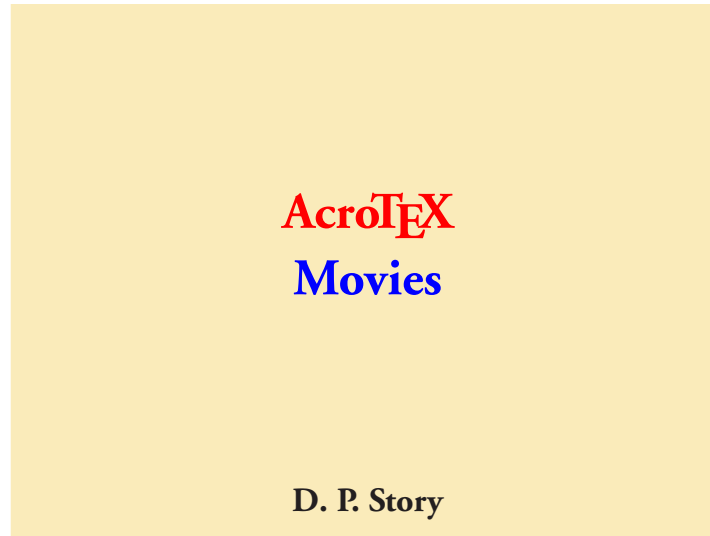
in the preamble of your document. You may use only one of the files `VideoPlayer.swf`, `VideoPlayerPlus.swf`, and `VideoPlayerX.swf` in a `rmannot` document.<sup>1</sup>

The RMA on the next page uses `VideoPlayerPlus.swf` to play any of three FLV files, two of them are embedded in this document, the third (the one about elephants) is streamed by an http server.

---

<sup>1</sup>It is possible to use more than one of these files in a document, but one should be sufficient.

Press any of the buttons below the RMA to play aFLV, note that the skins differ from video to video, and the color of the skins are different as well. The preamble of this document contains the usual declarations for an rmannot document, download the source file, [rma\\_vp-plus\\_pub.tex](#), for this RMA attached to this document.



The code and discussion follow:

```

1 \rmAnnot[name=playerPlus,poster=aebmovie_poster,
2   resources={sample,skin1,skin2,skin3}]{320bp}{240bp}{horse1}}\{3bp}
3 \pushButton[\TU{Embedded}\CA{Horse}\S{S}\A{\JS{%
4   \playJS{\Name{horse1}}{skin1}{0x5F5F5F}}]{playHorse}}{11bp}\kern1bp
5 \pushButton[\TU{Embedded}\CA{Playing}\S{S}\A{\JS{%
6   \playJS{\Name{sample}}{skin3}{0xFF0000}}]{playSample}}{11bp}\kern1bp
7 \pushButton[\TU{Streamed}\CA{Elephants}\S{S}\A{\JS{%
8   \playJS{\urlName{elephants}}{skin2}{0x0000FF}}]{playElephants}}{11bp}

```

#### Comments on the RMA code:

- Line 1: We give our RMA a name (name=playerPlus) so it can be referenced conveniently by the buttons.
- Line 2: The last argument of the \rmAnnot command requires the name of theFLV that will initially load in the RMA, here, this FLV is named horse1; horse1 is defined in the preamble

```

\newcommand{\myRMFiles}{%
  C:/Users/Public/Documents/My TeX Files/%
  tex/latex/aeb/aebpro/rmannot/RMfiles}
\saveNamedPath{horse1}{\myRMFiles/horse1.flv}

```

In the optional parameters, we list our resources. We plan to play the video named sample, defined in the preamble by

```
\saveNamedPath{sample}{\myRMFiles/sample.flv}
```

so we list `sample` in the resources list. This inclusion is necessary when the FLV is to be embedded. We also plan to use `skin1`, `skin2`, and `skin3`, so we list these as part of the resources: In line 2, observe the specification,

```
resources={sample,skin1,skin2,skin3}
```

We also plan to play the FLV named `elephants`, but we do not list it in the resources because this file is not embedded, it is to be streamed by the `www.math.uakron.edu` server.

Now let's look at the buttons; we discuss the second and third (the former one plays an embedded FLV, the latter plays an FLV on the Internet). The code for the second button is

```
1 \pushButton[\TU{Embedded}\CA{Playing}\S{S}\A{\JS{%
2   \playJS{\Name{sample}}{skin2}{0xFF0000}}]{playSample}{\kern1bp}
```

#### Comments on the Playing Button:

- Line 1: Standard key-value pairs for setting a tooltip (`\TU{Embedded}`), the button caption (`\CA{Playing}`), and the style of the button (`\S{S}`, `solid`). Lastly, we begin the key that sets the (JavaScript) action of the button (`\A{\JS}`). The code itself continues on line (2).
- Line 2: We define the JavaScript action using the  $\LaTeX$  command `\playJS`, defined in the preamble. This definition is

```
\newcommand{\playJS}[3]{%
  playPlusX("playerPlus", "#1", "\Name{#2}", "#3")}
```

`\playJS` is a simple command for populating the `playPlusX` JavaScript function, also defined in this document. The command takes three parameters:

- #1 is the *file name or url* of the FLV to be played. You enter the file name when the FLV is embedded. Refer to line (2) in the code for the button, we want to play the FLV whose symbolic name is `sample` so we use the convenience macro `\Name` to pass file name of the FLV. `\Name{sample}` expands to `sample.flv` (which can also be entered directly as the #1 parameter).
- #2 is the *symbolic name* for the skin to be used. In line (2) of the button code above, we pass `skin3` as #2. In the definition of `\playJS`, #2 becomes the argument of `\Name` command, `\Name{#2}`.
- #3 is the *color* of the skin, written in the form of `0xRRGGBB`.

Before discussing the `playPlusX()` function, we briefly discuss the Elephants button.

```
1 \pushButton[\TU{Streamed}\CA{Elephants}\S{S}\A{\JS{%
2   \playJS{\urlName{elephants}}{skin2}{0x0000FF}}]{playElephants}{\kern1bp}
```

#### Comments on the Elephants Button:

- Line 2: Again we populate the parameters of the `\playJS` macro. We want to play the FLV whose symbolic name is `elephants`. This symbolic name is defined in the preamble by

```
\definePath{\myHP}{http://www.math.uakron.edu/~dpstory}
\saveNamedPath{elephants}{\myHP/videos/elephants.flv}
```

The first parameter of `\playJS` is passed as an url with `\urlName{elephants}`.<sup>2</sup> The `\urlName` macro, defined in `rmannot`, expands to the full url of the symbolic name that is its argument.

The second parameter is `skin2`, and the third parameter is the color `0x0000FF` (blue).

## 2.1. The `playPlusX()` Function

The function `playPlusX()` is defined in the preamble of this document, it is not part of the `rmannot` package. It is a JS function designed to play the `VideoPlayerPlus.swf` and the `VideoPlayerX.swf` widgets.

```
1 function playPlusX(name,source,skin,skinColor) {
2     var rm=this.getAnnotRichMedia(this.pageNum,name);
3     rm.activated = true;
4     if (skin!="noChange") rm.callAS(\mmSkin, skin);
5     if (skinColor!="noChange") rm.callAS(\mmSkinColor, skinColor);
6     rm.callAS(\mmSource, source);
7     rm.callAS(\mmPlay);
8 }
```

The function takes four parameters

1. `name`: The name of the target Rich Media annotation. When using `\rmAnnot`, a name can be assigned using `name` key in the optional arguments list.
2. `source`: The source file to be played. The parameter `source` can be the name of a FLV file that has been embedded in the document, or a url of a FLV on the Web.
3. `skin`: The name of the skin to be used. The skin is typically embedded in the document, and is listed among the resources, or the value of the `skins` key.
4. `skinColor`: The skin color, in the form of `0xRRGGBB`.

**noChange**: A special comment is needed for lines (4) and (5). If you pass the string `noChange` as the `skin` (`skinColor`), line (4) (line (5)) is not executed. This is useful for creating general purpose play function that will play a source file without changing the skin or skin color.

The commands `\mmSkin`, `\mmSkinColor`, `\mmSource`, and `\mmPlay` are documented in [rmannot manual](#). Basically these are helper commands for executing the appropriate ActionScript function defined in `VideoPlayerPlus.swf`. The two widgets (Plus and X) have common functionality but the names of the function differ. The helper commands expand to the correct string depending on which video player is used.

<sup>2</sup>If we wanted to embed the elephants FLV in the document, we would use `\Name{elephants}` and list elephants in the resources of the `\rmAnnot` command. We would also have to change the definition of `elephants` to point to the file on the local hard drive, rather than on the Internet.

## 2.2. Other Variations

In this section, we look at a couple of variations on the code used earlier.

**Main file embedded, the other two on the Web.** To reduce the file size of the document, more FLVs may be put on the Web. Below is an example of the horse movie still embedded, but the the other two on the Web. In short, we want to move the sample video to the Web.

First, we must change the definition of `sample` from

```
\saveNamedPath{sample}{\myRMFiles/sample.flv}
```

to

```
\saveNamedPath{sample}{\myHP/videos/sample.flv}
```

Then, we change the code a little for the sample FLV.

```
1 \rmAnnot[name=playerPlus,poster=aebmovie_poster,
2   resources={skin1,skin2,skin3}{320bp}{240bp}{horse1}}\{3bp}
3 \pushButton[\TU{Embedded}\CA{Horse}\S{S}\A{\JS{%
4   \playJS{\Name{horse1}}{skin1}{0x5F5F5F}}]{playHorse}}{11bp}\kern1bp
5 \pushButton[\TU{Streamed}\CA{Playing}\S{S}\A{\JS{%
6   \playJS{\urlName{sample}}{skin3}{0xFF0000}}]{playSample}}{11bp}\kern1bp
7 \pushButton[\TU{Streamed}\CA{Elephants}\S{S}\A{\JS{%
8   \playJS{\urlName{elephants}}{skin2}{0x0000FF}}]{playElephants}}{11bp}
```

Only two changes were made in the above code. In line (2), `sample` as remove from the `resources`, it is no longer embedded; and in line (6) `\Name{sample}` was changed to `\urlName{sample}`

**All FLVs are on the Web.** To reduce the the file size even further, we could put all three files on the Web.

First, we must change the definition of `horse1` from

```
\saveNamedPath{horse1}{\myRMFiles/horse1.flv}
```

to

```
\saveNamedPath{horse1}{\myHP/videos/horse1.flv}
```

Of course we redefine of `horse1` as described above as well.

The code then becomes

```
1 \rmAnnot[url,name=playerPlus,poster=aebmovie_poster,
2   resources={skin1,skin2,skin3}{320bp}{240bp}{horse1}}\{3bp}
3 \pushButton[\TU{Streamed}\CA{Horse}\S{S}\A{\JS{%
4   \playJS{\urlName{horse1}}{skin1}{0x5F5F5F}}]{playHorse}}{11bp}\kern1bp
```

```

5 \pushButton[\TU{Streamed}\CA{Playing}\S{S}\A{\JS{%
6   \playJS{\urlName{sample}}{skin3}{0xFF0000}}]{playSample}{11bp}\kern1bp
7 \pushButton[\TU{Streamed}\CA{Elephants}\S{S}\A{\JS{%
8   \playJS{\urlName{elephants}}{skin2}{0x0000FF}}]{playElephants}{11bp}

```

The main file is `horse1` (by that I mean the FLV that is the last argument of the `\rmAnnot` command), is on the Web; we inform `\rmAnnot` of this by putting in the key `url` in the option list in line (1) (otherwise, `\rmAnnot` assumes the file is embedded; eventually there will be an error when the file is distilled because distiller cannot find the resource for embedding).

In line (6), we changed `\Name{horse1}` to `\urlName{horse1}`.

**All three FLVs are embedded.** To “fatten” the file size, we can embed all three files.

In the preamble, we must have

```

\newcommand{\myRMFiles}{%
  C:/Users/Public/Documents/My TeX Files/%
  tex/latex/aeb/aebpro/rmannot/RMfiles}
\saveNamedPath{horse1}{\myRMFiles/horse1.flv}
\saveNamedPath{sample}{\myRMFiles/sample.flv}
\saveNamedPath{elephants}{\myRMFiles/elephants.flv}

```

so all definitions point to files on the local hard drive.

The code should be

```

1 \rmAnnot[name=playerPlus,poster=aebmovie_poster,
2   resources={sample,elephants,skin1,skin2,skin3}
3 ]{320bp}{240bp}{horse1}\[3bp]
4 \pushButton[\TU{Embedded}\CA{Horse}\S{S}\A{\JS{%
5   \playJS{\Name{horse1}}{skin1}{0x5F5F5F}}]{playHorse}{11bp}\kern1bp
6 \pushButton[\TU{Embedded}\CA{Playing}\S{S}\A{\JS{%
7   \playJS{\Name{sample}}{skin3}{0xFF0000}}]{playSample}{11bp}\kern1bp
8 \pushButton[\TU{Embedded}\CA{Elephants}\S{S}\A{\JS{%
9   \playJS{\Name{elephants}}{skin2}{0x0000FF}}]{playElephants}{11bp}

```

Notice that in line (2) both `sample` and `elephants` are listed as resources. It is not necessary to list `horse1` as a resource, for in the absence of the `url` key in the optional parameter list, `\rmAnnot` automatically embeds the main file.

In lines(5), (7), and (9), `\urlName` is replaced by `\Name`.

Now, I simply must get back to my retirement! 