

AcroTeX.Net

**The rmannot Package**  
**Controlling the position of a Floating**  
**Window**

**D. P. Story**

## Introduction

One of the big features of **Acrobat 9** (and **Adobe Reader 9**) was the *rich media annotation* (RMA). I have written extensively on this subject in [AcroTeX Blog](#). A rich media annotation play Flash content (SWF, FLV, MP3). The RMA is a truly wonderful feature to Acrobat.

Some of the features of RMA, as documented in *Adobe Supplement to ISO 32000, BaseVersion 1.7, ExtensionLevel 3*, were not supported in Version 9. When Version 9 rolled out, the control of the RMA floating window was not support, though the PDF objects for this control is laid out in the *Adobe Supplement*.

This particular problem was corrected, to some extent, by **Acrobat X**. **Acrobat X** now obeys the `RichMediaPosition` dictionary (an internal PDF object), but there is no user interface (through Acrobat) to set the parameters. The [Acrobat Ninja](#) blog shows how to set the position of the floating window by editing the PDF file using a binary editor.

Users of the `rmannot` package do not have to edit the PDF file, nor wait for the next version of **Acrobat** in hopes that user interface is there. From the very beginning of Acrobat X, `rmannot` has supported the positioning of the floating window. With `rmannot`, the parameters of the `RichMediaPosition` can be specified.

## 1. The Floating Window Parameters

When the `windowed` key is set to `true`, the rich media annotation appears in a floating window. Use the `\setWindowDimPos` command to set the dimensions of the window and its positioning.

```
\setWindowDimPos{<key-value pairs>}
```

**Command Location:** This command may be placed anywhere and will take affect for the next rich media annotation created by `\rmAnnot`.

**Parameter Description:** There are a number of key-value pairs for setting the floating window; the default values are normally adequate for most applications.

- **width:** The width is described by three *key-value pairs*, `default`, `max`, and `min`, measured in default user space units. Default values: `default: 288, max:576, min: 72`.  
For example, `width={default=300,max=600,min=80}`.
- **height:** The height is described by three *key-value pairs*, `default`, `max`, and `min`, measured in default user space units. Default values: `default: 216, max:432, min: 72`.  
For example, `height={default=300,max=600,min=80}`.
- **position:** The position of the floating window is described by four key-value pairs.

- `halign`: The `halign` parameter describes the horizontal alignment of the window. Valid values are `near`, `center` and `far`. The default is `far`. For languages that read from left-to-right, a value of `near` refers to the left edge of the viewing window; whereas `far` refers to the right edge of the viewing window. (For right-to-left reading languages, the description of `near` and `far` are reversed.)
- `valign`: The `valign` parameter describes the vertical alignment of the window. Valid values are `near`, `center` and `far`. The default is `near`.
- `hoffset`: The description of `hoffset` is paraphrased from the *Adobe Supplement* document: The offset from the alignment point specified by the `halign` key. A positive value for `hoffset`, when `halign` is either `near` or `center`, offsets the position towards the `far` direction. A positive value for `hoffset`, when `halign` is `far`, offsets the position towards the `near` direction. The default is 18.
- `voffset`: The description of `voffset` is paraphrased from the *Adobe Supplement* document: The offset from the alignment point specified by the `valign` key. A positive value for `voffset`, when `valign` is either `near` or `center`, offsets the position towards the `far` direction. A positive value for `voffset`, when `valign` is `far`, offsets the position towards the `near` direction. The default is 18.

The `\resetWindowDimPos` command can be used to reset the floating window parameters to their default values.

```
\resetWindowDimPos
```



The default positioning of a floating window is the upper right corner of the Acrobat window, offset 18 points from the top and 18 points from the right edge. Click on the icon to the left to see the default positioning of a floating window. For **Acrobat 9**, the floating window is hard-wired to appear initially in this position.

In layman's terms the combination of `halign=far`, `valign=near` puts the floating window in the upper right corner of the active window of Adobe Reader/Acrobat, assuming a left-to-right reading language. The values of `voffset=18`, `hoffset=18`, moves the floating window 18 points down and 18 points to the left. That would be its initial position.

## 2. Positioning the Floating Window



One of the problems with the default positioning of a floating window is the resizing of the window. The control for resizing is in the lower-right corner of the window. To resize the window, you first have to reposition the window by dragging the title bar further from the right edge. A better choice would have been to place the window in the upper-left corner. Of course, you can only see the upper-left corner positioning if you are using Version 10 (**Acrobat X** or **Adobe Reader X**)

Now that the window is in the upper-left corner, the user, that's you, can easily drag the lower-right corner to resize the window.

### Centering the Window

We can also center the window by using the appropriate `position` parameters,

```
position={halign=center, valign=center}
```

The movie shown in the left margin is centered on the Acrobat (Adobe Reader) window.

Unfortunately, the position of the window cannot be dynamically set using JavaScript API, perhaps that will come at some later time.



### Placing the Window at the Bottom of the Application Window

One last example on this topic, we position the movie in the lower-right corner

```
position={halign=far, valign=far}
```

In this example, the floating window of the **RMA** in the left margin appears in the lower-right corner of the Acrobat (Adobe Reader) window.

Note that as you resize the application window, the **RMA** floating window will reposition itself to the lower-left corner.



## 3. Changing the offset

You may have noticed that we did not change the value of either `hoffset` or `voffset`; they remained at their default value of 18 (points). All the examples that appeared on one of the corners were properly offset 18 points from the edge of the application. In this section we'll adjust the offsets to demonstrate their use.

### Upper-left corner, 0 points offset



Let's place the floating window snugly in the corner with no offset to see how it looks. Click on the movie icon, click on it now!

### Centered vertically, offset to right



Now we want to center the window vertically within the application window, but move it horizontally to the far position, but offset it 72 points from the right edge of the application window. Again, click on the movie icon in the left margin and stand amazed! That's what I'm talking about. When the `halign`

parameter is `far`, the `hoffset` parameter offsets the floating window in the direction of `near` (to the left).

## 4. The size parameters



There are two size parameters, `width` and `height`, each of these take key-value pairs, the keys are `default`, `max`, and `min`. The `default` parameter seems to be the one to set the window size; it is not clear, from experimentation, whether the other two have any effect, they may not be implemented. Here, we set the window at 612 points by 612 points. The FLV will resize itself to fit within the window.

## 5. Some comments on bugs

As noted earlier, the `max` and `min` parameters seem to have no effect. These two  $\text{\TeX}$  parameters correspond, respectively, to the **Max** and **Min** keys in the **RichMediaWindow** dictionary.

Another point, the documentation in *Adobe Supplement to ISO 32000, BaseVersion 1.7, ExtensionLevel 3* states that the values of these and other parameters are in *default user space units*. The default user space unit is roughly  $1/72$  of an inch, or one Adobe point, or one  $\text{\TeX}$  big point (bp). However, many of these rich media parameters appear to be in *pixel measurement*.

Consider the last example above. The default width and height are set to 612 (points), which is 8.5 inches. If you click on the movie icon above, and put the magnification to the document at 100%, you'll see that the width of the floating window is not equal to the width of the page.



Contrast that window with this one. Maximize your window, put magnification on 100%, and click on the movie icon. You'll see the width of the movie (excluding the boundary) is (approximately) the width of the page. The default keys (PDF specification key **Default**) are set to 935 not 612. Where did 935 come from? Well on the Windows platform, the default resolution is 110 pixels/inch (this may be different than on a Mac platform). We made the following calculations, converting points to pixels

$$612 \text{ pts} = \frac{612 \text{ pts}}{72 \text{ pts/inch}} = 8.5 \text{ inches} = 8.5 \text{ inches} \cdot 110 \text{ pixels/inch} = 935 \text{ pixels}$$

It is this value, 935, that is used to set the width and height of the floating window. This suggests the measurement is in pixels.

I've also noticed that the offset keys don't like decimal numbers. If you set `hoffset=400`, you get a clear horizontal offset from center (say), but if you put `hoffset=400.1`, there is very little offset, probably the default value of 18.



D. P. Story

$\text{\TeX}$