

POWER ROTATOR - AS3 banner rotator - v.1.3



by Andrea Baroni - bluegestalt

user's manual

Hi, and **thank you** for having downloaded this software.

Files list:

- * PowerRotator.fla : Flash CS3 project file
- * PowerRotator.swf : Flash 9 player compiled
- * PowerRotator.xml : the xml file that defines the gallery and its images
- * [demo] : folder with an usage example
- * PowerRotator(withEmbeddedFonts).fla : Flash CS3 project file with embedded fonts (used in the demo)
- * [classes] : classes used in the project
- * Help.pdf: this doc

Fonts used:

- * Myriad Web (only as embedded font in the usage example inside the demo folder)

AS language version:

- * AS3 PowerRotator.xml

HOW TO MODIFY THE APPLICATION TO YOUR NEEDS:

This is a powerful yet easy to use and configure Flash AS3 **banner rotator**, which means that you can "rotate" (automatically or not) a gallery of images or swf inside its content.

You can **add the rotator** in your **html page**, using your preferred method, but I recommend the SWFObject method (*see usage example in demo folder*).

You can also **add the rotator** to **another AS3 Flash application**, using the *Loader class* to load it as an *external swf*.

*There are two **ready to use swf files**: one in the main folder, which size is 24Kb, and there is another one in the "demo" folder, which size is 52Kb, and that contains the embedded font "Myriad Web".*

Use the first one if you don't want to use embedded fonts, or use the second one if you wish to use embedded fonts (in this case the Myriad Web font), just like in the usage example.

Configure the xml, put the banners you want, write the captions, adjust the settings, etc...

You need to **open the fla and compile again the rotator swf** mainly when:

- * you need to **change the banner rotator size** (default is 590x300)
- * you need to **embed different fonts**

HTML FLASH OBJECT OPTIONS:

xml: defines the path of the xml file to load (default: *PowerRotator.xml*)

bannersFolder: defines the path of the folder containing the banners images and swf (always relative to the swf location). It OVERRIDES the xml option `banners_folder` and is used mainly to set a *dynamic path* for the banners from the html page (using for example php or asp).

After having embedded the power rotator swf in your page you just have to configure it, **editing the connected XML file.**

You can start from the empty (except the options) **PowerRotator.xml** in the **main folder**, or edit the **PowerRotator.xml** in the **demo folder**, which is already filled with the data of that usage example.

FORMATS SUPPORTED FOR THE BANNERS:

jpeg, png (with transparency), gif images and swf files.

CHANGING THE APPLICATION SIZE:

To **change the banner rotator size** you have to:

- * **open the fla**
- * in the *main properties* panel click on the **Edit...** button
- * **set** the desired **dimensions** for the application (*i.e.: 600x400*)
- * **compile again the swf**

NOTE: when you publish the application always remember to set the destination folder in the **publish settings**, or else the swf will be created in the folder which contains the fla.

Here you are... now the banners and the interface will adapt to the new size you set. Just remember to set also the *properties in the HTML* for the embedded flash object (SWFObject or anything else you have used).

SCARED BY ALL THE OPTIONS?

Keep it easy! If you just want a simple banner rotator you can always keep the *default options* (they are here for this!) and just add your banners.

Only if you want advanced features, or want to be able to control in depth each banner (like in the usage example) you need to touch all the different options.

EDITING THE XML:

OPTIONS:

These are the **default options** for the application:

```
<options
```

```
  application_maxWidth="0"  
  application_maxHeight="0"  
  application_transparentBackground="yes"  
  application_enableJavascriptCallbacks="no"
```

APPLICATION
OPTIONS

```
  banners_folder="banners"  
  banners_autoPlay="yes"  
  banners_delay="5"  
  banners_randomOrder="no"  
  banners_resize="fill"  
  banners_keepProportions="yes"  
  banners_alignMiddle="yes"  
  banners_transitionFx="1"  
  banners_transitionRandom="no"  
  banners_kenBurns="no"  
  banners_kenBurnsZoom="in"  
  banners_mask="no"  
  banners_overlay="no"  
  banners_overlayWidth="0"  
  banners_overlayHeight="0"  
  banners_overlayMarginLeft="0"  
  banners_overlayMarginTop="0"  
  banners_overlayTransparency="50"
```

ALL BANNERS
OPTIONS

```
  interface_buttonPrev="yes"  
  interface_buttonPrevPosition="right"  
  interface_buttonNext="yes"  
  interface_buttonNextPosition="right"  
  interface_buttonPause="yes"  
  interface_buttonPausePosition="right"  
  interface_timerVisible="yes"  
  interface_timerPosition="left"  
  interface_linkCursor="yes"  
  interface_autoHide="yes"
```

INTERFACE
OPTIONS

```
  captions_appearFromSide="bottom"  
  captions_autoHide="no"  
  captions_useEmbeddedFonts="yes"  
  captions_font="Arial"  
  captions_textAlign="left"  
  captions_textSize="15"  
  captions_lineSpacing="2"  
  captions_textColor="FFFFFF"  
  captions_linksColor="FF0000"  
  captions_textShadow="no"  
  captions_backgroundColor="000000"  
  captions_backgroundTransparency="70"  
  captions_paddingBottom="0"  
  captions_paddingLeftRight="0"
```

ALL CAPTIONS
OPTIONS

```
/>
```

application_maxWidth *[integer]* : the application will always **resize to fit the entire stage**. With this value you can set a maximum width in pixels for the banner rotator (the application will be placed in (0,0)). **Set to 0 to have no limit to the width.**

application_maxHeight *[integer]* : the application will always **resize to fit the entire stage**. With this value you can set a maximum height in pixels for the banner rotator (the application will be placed in (0,0)). **Set to 0 to have no limit to the height.**

application_transparentBackground *[yes/true | no/false]* : if set to transparent (yes or true) the application will use no background. If set to "no" a default gray gradient will be placed under the banners. Set this option to yes and use **wmode="transparent"** in the html to have the banner rotator background completely transparent (see *usage example in demo folder*).

application_enableJavascriptCallbacks *[yes/true | no/false]* : if set to yes the rotator will accept javascript callbacks, to control the rotator from an html page.

The valid **callbacks** are:

- **pausePlaySlideshow**
- **goNextBanner**
- **goPrevBanner**
- **goBanner(id)**

To see how to **setup the javascript side** please see *usage example* in the demo folder.

banners_folder : define the path where the banners media (jpg, gif, swf, etc...) are contained. **Always use a relative (to the rotator swf) path** . If you need to use a dynamic path (php, asp, etc...) please use the **bannersFolder flash object option** in the html.

banners_autoPlay *[yes/true | no/false]* : if set to true the banners' sequence will play automatically. If set to false the user must click on the NEXT or PREVIOUS BUTTON to change banner.

banners_delay : defines the default time in seconds for each banner, used when autoPlay is set to true [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

banners_randomOrder *[yes/true | no/false]* : if set to true the banners' sequence will always play random and not as defined in the xml.

banners_resize *[fill | fit | no/false]* : if set to "fill" the banners will be resized to fill the rotator swf dimensions. If set to "fit" the banners will be resized to be all visible inside the rotator. If set to "no" (or false) no resize will take place (maintain original dimensions) [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

banners_keepProportions *[yes/true | no/false]* : if set to true the banners will be resized always keeping the file original proportions, so without deforming it. Used only when `resizeToFit` is set to true. [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

banners_alignToMiddle *[yes/true | no/false]* : if set to true the banners will be always align vertically and horizontally to the middle of the movie. If set to false the banners will be placed in the top left corner (0,0) [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

banners_transitionFx [0 to 10] : defines the transition effect used to switch from a banner to another. Use the *table below* to put the correct number for the desired effect. [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

IMMEDIATE = 0	SCALE OUT = 6
CROSSFADE = 1;	BLINDS = 7
FADE FROM BLACK = 2	WINDOWS = 8
FLASH = 3	FADE OUT FADE IN = 9
SLIDE HORIZONTALLY = 4	FLIP = 10
SLIDE VERTICALLY = 5	

banners_transitionRandom [yes/true | no/false] : if set to true the banners always use a random transition (chosen from 1 to 8). This overrides every other settings on transitions.

banners_kenBurns [yes/true | no/false] : if set to true the banners will use a Ken Burns like effect once displayed. [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

banners_kenBurnsZoom [in | out | random] : when using the ken burns effect you can define if it will use a zoom in or zoom out effect (or choose randomly between these two). [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

banners_mask [no/false | (external swf path)] : you can load an external swf and use it as a **vector mask** for the whole rotator. The mask can have animations (*see usage example and masks examples*). The masks external swf files must be palced **inside the banners folder**. If set to "no" (or false) no mask will be used for the banners. [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

banners_overlay [no/false | (external image or swf path)] : you can load an external image (jpg, gif or png) or swf file, that will be placed as an overlay layer, above the banners and beneath the interface. This layer will remain fixed in all the banners. If set to "no" (or false) overlay will be placed. (*see usage example in demo folder*).

banners_overlayWidth [number] : you can specify the overlay layer width. If set to 0 the original dimensions will be mantained.

banners_overlayHeigth [number] : you can specify the overlay layer height. If set to 0 the original dimensions will be mantained.

banners_overlayMarginLeft [number] : you can specify the left margin of the overlay layer.

banners_overlayMarginTop [number] : you can specify the top margin of the overlay layer.

banners_overlayTransparency [0 to 100] : you can specify the alpha value (transparency) for the overlay layer.

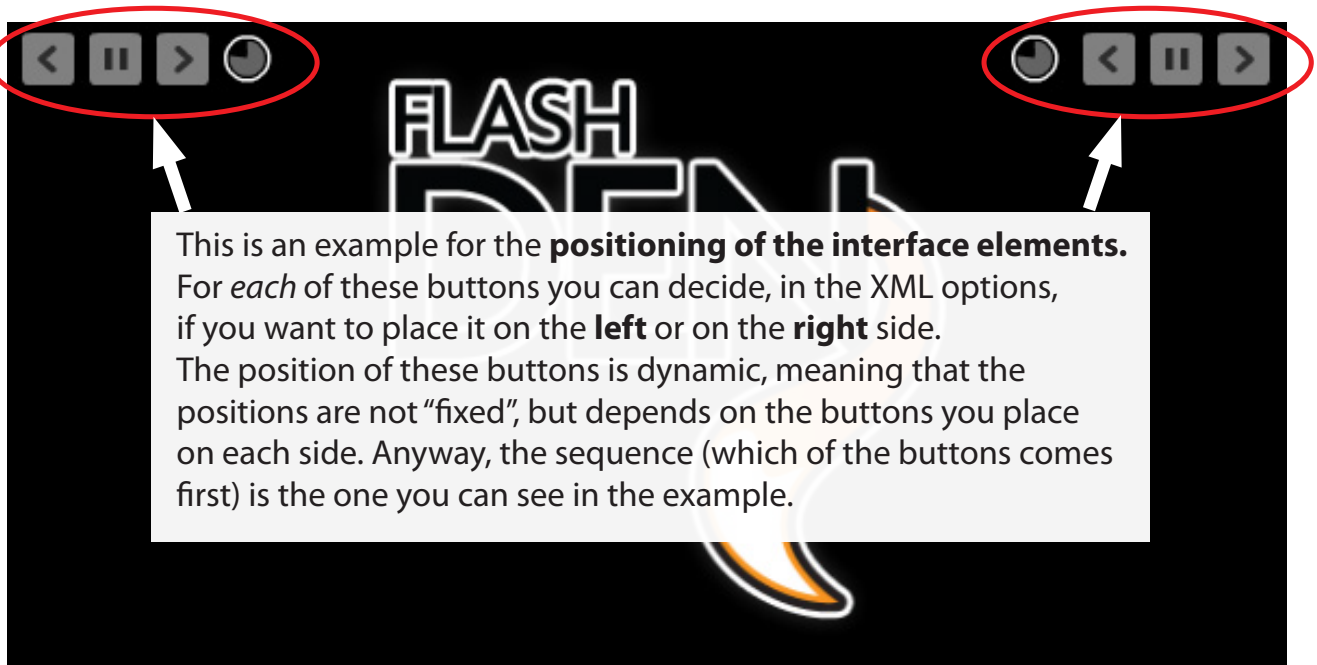


figure 1

interface_buttonPrev [yes/true | no/false] : if set to true the “go previous banner ” button will be added to the interface and displayed (*default position*: top right)

interface_buttonPrevPosition [left | right] : you can place this button to the top left or the top right of the application (*see figure 1*).

interface_buttonNext [yes/true | no/false] : if set to true the “go next banner ” button will be added to the interface and displayed (*default position*: top right)

interface_buttonNextPosition [left | right] : you can place this button to the top left or the top right of the application (*see figure 1*).

interface_buttonPause [yes/true | no/false] : if set to true the “pause/play ” button will be added to the interface and displayed (*default position*: top right)

interface_buttonPausePosition [left | right] : you can place this button to the top left or the top right of the application (*see figure 1*).

interface_timerVisible [yes/true | no/false] : if set to true a little timer clock will be displayed in the top left corner of the application (showing the remaining time for that banner). Used if `autoplay` is set to true.

interface_timerPosition [left | right] : you can place the timer to the top left or the top right of the application (*see figure 1*).

interface_linkCursor [yes/true | no/false] : if you define a **link** for a banner, clicking on the image will make the browser go to that url. If this options is set to true a *small cursor animation* will remaind you that there’s a link present for that banner.

interface_autoHide [yes/true | no/false] : if set to true when the mouse leave the application the interface (buttons and timer) will hide in fade, and return automatically as the mouse enter again the application.

captions_appearFromSide [*bottom | top | left | right | random | fixed/no*]: defines the direction of appearance of the caption. It can come from one of the four sides, choose randomly between these four values, or be fixed with the whole banner (set to “fixed” or “no”) [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

captions_autoHide [*yes/true | no/false*]: if set to true the captions will hide as the mouse will leave the application, and return automatically as the mouse enter again the application. [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

captions_useEmbeddedFont [*yes/true | no/false*]: if set to true the banners will by default use an embedded font. Remember that YOU NEED TO EMBED THAT FONT ALSO IN THE SWF (*see later*). [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

captions_font: the default font used in the banners captions. Remember to write it correctly or the application will not be able to locate it. IF YOU ARE USING AN EMBEDDED FONT HERE REMEMBER TO SET “useEmbeddedFont” TO TRUE, AND EMBED THAT FONT ALSO IN THE SWF (*see later*). [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

captions_textAlign [*left | center | right*]: the default align setting for the captions texts. [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

captions_textSize [*number*]: the default size for the captions texts. [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

captions_lineSpacing [*number*]: the default paragraph line spacing for the captions texts. [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

captions_textColor [*RGB value*]: the default color for the captions texts. [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

captions_linksColor [*RGB value*]: the default color for the html links in the captions texts. [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

captions_textShadow [*yes/true | no/false*]: if set to true the caption text will have a soft shadow [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

captions_backgroundColor [*RGB value*]: the default color for the banners captions background. [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

captions_backgroundTransparency [*0 to 100*]: the default alpha value (transparency) for the banners captions background. [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

captions_paddingBottom [*number*]: defines the margin from the bottom of the rotator of the caption text. [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

captions_paddingLeftRight [*number*]: defines the margin from the left and right sides of the rotator of the caption text. [\[POSSIBILITY TO OVERRIDE THIS IN THE SINGLE BANNER OPTIONS\]](#)

BANNERS:

These is how you can add and define a single banner.

Just add a BANNER NODE in the XML:

```
<banner
  file="yourFileName"
  link="your link url"
  target="_blank">
  <![CDATA[Write the text of your caption here...]]>
</banner>
```

Only the **file** attribute is *mandatory*.

Remember that if you need to use **HTML formatting** inside your caption you need to put your text inside the “<![CDATA[” and “]]>” strings.

Valid HTML formatting: <p>, </p>,
, , , , , <i>, </i>, , , <a>, , (for size and color), , , , ,

BANNERS OPTIONS:

These are the banners options you can *optionally* set:

```
<banner
  ...
  transitionIn=...           [0 to 8]
  delay=...
  resize=...                 [fill | fit | no/false]
  keepProportions=...       [yes/true | no/false]
  alignMiddle=...           [yes/true | no/false]
  kenBurns=...               [yes/true | no/false]
  kenBurnsZoom=...          [in | out | random]
  mask=...                   [no/false | external swf path]
  font=...
  captionAppearFromSide=... [bottom | top | left | right | random | fixed/no]
  captionAutoHide...        [yes/true | no/false]
  useEmbeddedFont=...       [yes/true | no/false]
  textSize=...
  textAlign=...              [left | center | right]
  lineSpacing=...
  textColor=...              [RGB value as RRGGBB]
  linkColor=...              [RGB value as RRGGBB]
  textShadow=...            [yes/true | no/false]
  backgroundColor=...       [RGB value as RRGGBB]
  backgroundTransparency=... [0 to 100]
  paddingBottom=...
  paddingLeftRight=...
  swfStartingFrameLabel=... [a frame label for the external swf]
  ...
</banner>
```

These **overrides the global options settings**, so please refer to the options paragraph to understand what they will do.

EMBEDDING THE FONTS:

To use **embedded fonts** with the captions you have to:

- * set the `captions_useEmbeddedFonts` to *yes* or *true* in the **options**, which makes the application use embedded fonts in *all the banners*, or alternatively set the `useEmbeddedFont` **banner option** just for the banners where you intend to use them.
- * set the font name in `captions_font` **options** (global setting) or `font` **banner option** (single banner setting).
- * **include the fonts in the swf**. To do this:
 - **open the fla**: there's a fla called **PowerRotator(withEmbeddedFonts)**, which already include a font (the one used in the usage example), so you could start with this.
 - once opened you should be in the **"fonts" scene** of the movie. If not open it.
 - under the main movie area you will see **3 dynamic textfields**: in this example they are all the same font, but one is for the *regular* option, one for the *bold*, and one for the *italic*. Remember that if you want to use an embedded font with all these *different styles* you have to include all of them.
 - here you can **change the font value** to the font you want.
 - remember to click on the button **CHARACTER EMBEDDING** in the properties panel, and set the characters you want to embed. DON'T USE "ALL", just put the characters you will use.
 - **comple again the swf** with your fonts. If you want to include *more fonts* just insert more dynamic text fields with different fonts and/or styles.

CREATING EXTERNAL MASKS:

In the "demo" folder you can find a folder called **"masks examples"**, which contains the fla of the two masks loaded in the usage example.

Please open these two files to understand how to make a mask for the rotator. Once compiled to swf remember to put them in the "banners" folder, as set in the XML.

GENERAL GUIDELINES:

- The masks will be **vector masks**, meaning that no alpha will be taken in consideration
- Put everything in the main timeline and starting from frame 1
- You can have animations (*see example 1*)
- Don't overlap layers
- The masks will be stretched to fit the rotator size, but you will obtain the best results when you create masks that are the exact dimension of the banner rotator.

CONTROLLING EXTERNAL SWF TIMELINE ANIMATIONS:

When an external swf is loaded as a banner, its *timeline* will *start automatically*, even before that it is added to the stage (this is a Flash issue related to the Loader).

But you can put a `stop()` in your main timeline, where you want the swf to be, once it appears.

Now you can also set a `swfStartingFrameLabel` in the XML, for each banner node. This will define a **frame label** that will be called (with a `gotoAndPlay(label)` function) once the external swf has appeared in the rotator, but only after that its transition has ended. When that banner will leaves the stage its timeline will be restarted (from frame 1), to make this function works again the next time that the banner will appear.

If that option is *not set* just a `gotoAndPlay(1)` function will be called after the transition, making the timeline to restart from the first frame.

Please look at the *usage example* in the demo folder to see it in action, and open the "*swfExample.fla*" file in the **external swf examples** folder to see how to create an swf with this technique.

NOTE: you can use this technique only with **AS3 swf files**. You can load an AS2 swf as a banner, but you will not be able to control its timeline.

if you have any questions or problems please **contact me**