



Screen Banner Rotator version 1.2.0

ADVANCED COMPONENT
FOR FLASH 8, FLASH CS3/CS4

1. Overview	2
2. How to use	3
3. XML file structure	5
4. ActionScript Documentation	11
5. Hints and Tips	17

Overview

The Screen Banner Rotator is an XML-based Flash banner rotator with extensive set of parameters and attractive transition effects.

Main features:

- supports JPG, PNG, GIF image formats, as well as SWF files;
- resizable to the extent of 1920 x 1200 pixels;
- 3 different transition effects (can be chosen separately, in any combination);
- separate delay time for each image can be set (optionally);
- sequential or random order of images and/or transitions;
- images of any size can be used (they will be automatically resized for the best fit);
- adjustable transition duration (for the “fade” transition);
- fully adjustable controls and description text area;
- the font can be embedded or not;
- supports HTML formatted descriptions;
- FlashVars support for different XML files loading;
- once the current image is loaded and displayed, the next image can be loaded in background so the preloader would appear as least as possible.

You can work with the Screen Banner Rotator in one of the following ways:

1. Using the source FLA file (editable Flash file) for further publishing (creating SWF file).
2. Using the SWF file (fully working Flash movie). In this case, you don't need Flash software or any programming knowledge.

How to use

You can fully use the Screen Banner Rotator by editing the provided XML file using any text editor (such as Notepad for example). This will allow you to edit all the parameters.

You can create an application with the Screen Banner Rotator in one of the following ways:

1. Using the source FLA file (editable Flash file) for further publishing (creating SWF file)

In this case, you will need Macromedia Flash 8 or Adobe Flash CS3/CS4 software. The following files are required (included in the package):

- ScreenBannerRotator.fla;
- ScreenBannerRotator.as;
- TweenRGB.as;
- settings.xml.

Edit the parameters in the XML file (settings.xml) according to your needs. Open the source FLA file (ScreenBannerRotator.fla) and publish the Flash document with appropriate name.

2. Copying the Screen Banner Rotator library assets from the source FLA file into another document

In this case, you will need Macromedia Flash 8 or Adobe Flash CS3/CS4 software. The following files are required (included in the package):

- ScreenBannerRotator.fla;
- ScreenBannerRotator.as;
- TweenRGB.as;
- settings.xml.

Edit the parameters in the XML file (settings.xml) according to your needs. Open the source FLA file (ScreenBannerRotator.fla) and the destination FLA file. Drag the "Screen Banner Rotator" folder from the source document library onto the Stage of the destination document. Delete the copied assets from the Stage. Select the "ScreenBannerRotator" symbol (Movie Clip) in the Library panel in the destination document and drag it onto the Stage. When publishing the Flash document, note that the ActionScript files — ScreenBannerRotator.as and TweenRGB.as — must be in the same directory with the destination FLA file.

3. Using the SWF file (fully working Flash movie)

In this case, you don't need Flash software or any programming knowledge. The following files are required (included in the package):

- BannerRotator.swf;
- settings.xml.

Edit the parameters in the XML file (settings.xml) according to your needs.

If you need to change the name of the XML file, you can easily do it. Just open the ScreenBannerRotator.as file (before publishing the Flash document) and replace "settings.xml" with the name of your XML file. You can also specify the name for the XML file when embedding the Screen Banner Rotator into HTML page.

Below is the code that describes how to embed the Screen Banner Rotator using SWFObject 2:

```
<head>
  <script type="text/javascript" src="swfobject.js"></script>
  <script type="text/javascript">
    var flashvars = {xmlURL: "settings.xml"};
    var params = {};
    var attributes = {};
    swfobject.embedSWF("BannerRotator.swf", "bannerRotator", "450", "300", "8.0.0", false, flashvars,
      params, attributes);
  </script>
</head>
<body>
  <div id="bannerRotator">
    <a href="http://www.adobe.com/go/getflashplayer">
      
    </a>
  </div>
</body>
```

XML file structure

The basic template of the XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<rotator>

  <settings>
    <!-- General -->
    <bannerWidth value="450"/>
    <bannerHeight value="300"/>
    <bannerCornerRadius value="0"/>
    <randomImages value="false"/>
    <delayTime value="3"/>
    <autoPlay value="true"/>
    <pauseOnMouseOver value="false"/>
    <repeatMode value="true"/>
    <currentImageIndex value="0"/>
    <transparentImages value="true"/>

    <!-- Transitions -->
    <transitionType value="blinds,windows,fade"/>
    <randomTransitions value="true"/>
    <blindsEffectStripsNum value="10"/>
    <blindsEffectStripsColor value="006600"/>
    <blindsEffectStripsAlpha value="40"/>
    <windowsEffectSquaresSize value="90"/>
    <windowsEffectSquaresColor value="FF6600"/>
    <windowsEffectSquaresAlpha value="20"/>
    <windowsEffectGridColor value="006600"/>
    <windowsEffectGridAlpha value="60"/>
    <windowsEffectBlurFilter value="true"/>
    <fadeEffectDuration value="0.7"/>

    <!-- Controls -->
    <controlsHPosition value="right"/>
    <controlsVPosition value="center"/>
    <controlsOrientation value="vertical"/>
    <controlsMargin value="10"/>
    <showControls value="true"/>
    <showNumberButtons value="true"/>
    <showControlsScreen value="false"/>
    <controlsOrder value="playpause,timer"/>
    <controlsAutoHide value="false"/>
    <controlsScreenColor value="000000"/>
    <controlsScreenAlpha value="30"/>
    <controlsScreenPadding value="5"/>
    <controlsScreenStretchFull value="false"/>

    <!-- Buttons -->
    <buttonSpacing value="2"/>
```

```

<buttonBgColor value="000000"/>
<buttonBgAlpha value="50"/>
<buttonTextColor value="FFFFFF"/>
<buttonOverBgColor value="FFFFFF"/>
<buttonOverBgAlpha value="30"/>
<buttonOverTextColor value="000000"/>
<buttonSelectedBgColor value="FFFFFF"/>
<buttonSelectedBgAlpha value="70"/>
<buttonSelectedTextColor value="000000"/>
<buttonRollAnimationDuration value="0.1"/>

<!-- Description -->
<descriptionWidth value="600"/>
<descriptionHPosition value="center"/>
<descriptionVPosition value="bottom"/>
<showDescription value="true"/>
<textFadingDuration value="0.5"/>
<textSize value=""/>
<textColor value=""/>
<embedFonts value="true"/>
<textAlignment value="left"/>
<textLineSpacing value="0"/>
<textLetterSpacing value="0"/>
<textMarginLeft value="10"/>
<textMarginRight value="200"/>
<textMarginTopBottom value="5"/>
<textBgColor value="000000"/>
<textBgAlpha value="30"/>

<!-- Preloader -->
<preloaderColor value="FFFFFF"/>
<preloaderScale value="80"/>
<showPreloader value="true"/>
</settings>

<images>
  <image src="images/image1.jpg" url="" target="" delayTime=""><![CDATA[Description text for this
  image...]]></image>
</images>

</rotator>

```

The description of the XML file elements

1. The following table describes the nodes that the **<settings>** element contains:

Node name	Possible values	Default value	Description
GENERAL PROPERTIES			
bannerWidth	0-1920	600	The width of the banner, in pixels.

bannerHeight	0-1200	450	The height of the banner, in pixels.
bannerCornerRadius	≥0	0	The corner radius for the banner area, in pixels.
randomImages	true, false	false	Determines whether to display images in sequential (false) or random (true) order.
delayTime	≥0	5	Amount of delay between images, in seconds.
autoPlay	true, false	false	Specifies whether the banner should start rotating images automatically (true) or not (false).
pauseOnMouseOver	true, false	false	Specifies whether the banner should pause rotating images when the mouse is over it's area (true) or not (false).
repeatMode	true, false	true	Defines the banner behavior when it reaches the last image.
currentImageIndex		0	The index of the image to be displayed first in the sequence when the XML file is loaded. This is a zero-based index, so 0 is the first image.
transparentImages	true, false	false	Determines whether to use specific (true) or ordinary (false) mechanism for transitions. The specific one is needed if transparent images (PNG, GIF or SWF) are used.
TRANSITIONS PROPERTIES			
transitionType	fade, blinds, windows	fade, blinds, windows	The type of transition to be applied when rotating images. Multiple values should be separated by commas, without blank space after commas.
randomTransitions	true, false	true	Determines whether to apply transition types (if multiple values are selected) in sequential (false) or random (true) order.
blindsEffectStripsNum	1-40	14	The number of strips to be used for the "Blinds" transition effect.
blindsEffectStripsColor		FFFFFF	The color of strips for the "Blinds" transition effect.
blindsEffectStripsAlpha	0-100	40	The alpha transparency of strips for the "Blinds" transition effect.
windowsEffectSquaresSize		100	The size of squares for the "Windows" transition effect, in pixels. The minimum value is limited by the maximum allowed number of squares in a row or a column (10).

windowsEffectSquaresColor		FFFFFF	The color of squares for the "Windows" transition effect.
windowsEffectSquaresAlpha	0-100	40	The alpha transparency of squares for the "Windows" transition effect.
windowsEffectGridColor		FFFFFF	The color of grid lines for the "Windows" transition effect.
windowsEffectGridAlpha	0-100	40	The alpha transparency of grid lines for the "Windows" transition effect.
windowsEffectBlurFilter	true, false	true	Determines whether to apply the Blur Filter for the "Windows" transition effect.
fadeEffectDuration	0-3	0.8	The length of time that the "Fade" transition effect takes, in seconds.
CONTROLS PROPERTIES			
controlsHPosition	left, center, right	right	The horizontal position of the controls.
controlsVPosition	top, center, bottom	bottom	The vertical position of the controls.
controlsOrientation	horizontal, vertical	horizontal	The geometric orientation of the controls.
controlsMargin	≥0	10	The margins of the controls, relative to the nearest edges of the banner, in pixels.
showControls	true, false	true	Determines if the controls are visible.
showNumberButtons	true, false	true	Determines whether to show the number buttons in the controls (true) or not (false).
showControlsScreen	true, false	true	Determines whether to show the background (screen) of the controls (true) or not (false).
controlsOrder	playpause, previous, next, timer	undefined	Specifies which navigation buttons to show in the controls, and defines the order in which the buttons are displayed (separated by commas).
controlsAutoHide	true, false	false	Determines whether to show (hide) the controls automatically when the mouse is over (out of) the banner.
controlsScreenColor		FFFFFF	The color of the background (screen) of the controls.
controlsScreenAlpha	0-100	50	The alpha transparency of the background (screen) of the controls.
controlsScreenPadding	0-20	2	The spacing between the edges of the controls screen and the buttons block, in pixels.
controlsScreenStretchFull	true, false	false	Set this property to true if you want to stretch the controls screen to the width

BUTTONS PROPERTIES

buttonSpacing	≥0	2	(controlsOrientation="horizontal") or the height (controlsOrientation="vertical") of the banner. The value of controlsMargin in this case will be equal to 0.
buttonBgColor		000000	The background color of a button.
buttonBgAlpha	0-100	50	The background transparency of a button.
buttonTextColor		FFFFFF	The text color of a button's label.
buttonOverBgColor		FFFFFF	The background color of a rolled-over button.
buttonOverBgAlpha	0-100	80	The background transparency of a rolled-over button.
buttonOverTextColor		000000	The color of text when the pointer rolls over a button.
buttonSelectedBgColor		FFFFFF	The background color of a selected button.
buttonSelectedBgAlpha	0-100	80	The background transparency of a selected button.
buttonSelectedTextColor		000000	The text color of the selected button's label.
buttonRollAnimationDuration	0-0.3	0.1	The length of time that the animation takes when the pointer rolls over a button, in seconds.

DESCRIPTION PROPERTIES

descriptionWidth	≥0	bannerWidth	The width of the description area, in pixels.
descriptionHPosition	left, center, right	center	The horizontal position of the description area.
descriptionVPosition	top, bottom	bottom	The vertical position of the description area.
showDescription	true, false	true	Determines if the description area is visible.
textFadingDuration	0-0.3	0.3	The length of time that the text fading effect takes, in seconds.
textSize	8-96	11	The point size for the font of the description text.
textColor		FFFFFF	The color of the description text.
embedFonts	true, false	true	A Boolean value that indicates whether the font of the description text is an embedded font.

textAlignment	left, center, right, justify	left	The text alignment for the description.
textLineSpacing	-36...72	0	The line spacing for the description text, in points.
textLetterSpacing	-60...60	0	The letter spacing for the description text, in points.
textMarginLeft	≥0	10	The left margin of the description text, relative to the left edge of the description area, in pixels.
textMarginRight	≥0	10	The right margin of the description text, relative to the right edge of the description area, in pixels.
textMarginTopBottom	≥0	10	The top and bottom margins of the description text, relative, respectively, to the top and bottom edge of the description area, in pixels.
textBgColor		000000	The background color of the description area.
textBgAlpha	0-100	30	The background transparency of the description area.
PRELOADER PROPERTIES			
preloaderColor		FFFFFF	The color of the preloader.
preloaderScale	≥0	100	The scale of the preloader, in percent.
showPreloader	true, false	true	Determines whether to show the preloader (true) or not (false).

2. The **<images>** element can contain an unlimited number of **<image>** nodes. Each **<image>** node represents an image object and should have one required attribute `src`. You can also add 3 optional attributes in the **<image>** node: `url`, `target`, `delayTime`.

Node name	Attribute name	Default	Description
image	src	undefined	Specifies the URL of an image.
	url	undefined	(Optional) Specifies the URL of the page to be loaded when clicking on the image.
	target	_self	(Optional) Specifies the window or HTML frame that the page is loaded into ("_self", "_blank", "_parent", "_top" or your custom value).
	delayTime	undefined	(Optional) Specifies amount of delay which is to occur between this particular image and the next one, in seconds.

You can write a description text (optional) inside an **<image>** node. If you want to use standard HTML for the description text, you can put it inside a CDATA tag (see the basic XML template above).

ActionScript Documentation

Inheritance MovieClip > ScreenBannerRotator class

Methods unique to the ScreenBannerRotator class

The following table lists methods of the ScreenBannerRotator class.

Method	Description
ScreenBannerRotator.gotoImageAt()	Navigates to the image with the specified index.
ScreenBannerRotator.gotoNextImage()	Navigates to the next image in the gallery.
ScreenBannerRotator.gotoPreviousImage()	Navigates to the previous image in the gallery.

Properties unique to the ScreenBannerRotator class

The following table lists properties of the ScreenBannerRotator class.

Property	Description
ScreenBannerRotator.autoPlay	Specifies whether or not the banner should start rotating images automatically.
ScreenBannerRotator.controlsAutoHide	Determines whether to show (hide) the controls automatically when the mouse is over (out of) the banner.
ScreenBannerRotator.controlsOrder	Specifies which navigation buttons to show in the controls, and defines the order in which the buttons are displayed.
ScreenBannerRotator.delayTime	Amount of delay between images, in seconds.
ScreenBannerRotator.pauseOnMouseOver	Specifies whether or not the banner should pause rotating images when the mouse is over its area.
ScreenBannerRotator.randomImages	Determines whether to display images in sequential or random order.
ScreenBannerRotator.randomTransitions	Determines whether to apply transition types in sequential or random order.
ScreenBannerRotator.repeatMode	Defines the banner behavior when it reaches the last image.
ScreenBannerRotator.showControls	Determines if the controls are visible.
ScreenBannerRotator.showControlsScreen	Determines whether or not to show the background (screen) of the controls.
ScreenBannerRotator.showDescription	Determines if the description area is visible.
ScreenBannerRotator.showNumberButtons	Determines whether or not to show the number buttons in the controls.
ScreenBannerRotator.transitionType	The type of transition to be applied when rotating images.
ScreenBannerRotator.xmlFileURL	The path to the XML file.

ScreenBannerRotator.autoPlay

Usage

```
my_banner.autoPlay
```

Description

Property; specifies whether the banner should start rotating images automatically ("true") or not ("false"). The default value is "false".

Example

The following example sets the autoPlay property to "true":

```
my_banner.autoPlay = true;
```

ScreenBannerRotator.controlsAutoHide

Usage

```
my_banner.controlsAutoHide
```

Description

Property; determines whether to show (hide) the controls automatically when the mouse is over (out of) the banner. The default value is "false".

Example

The following example sets the controlsAutoHide property to "true":

```
my_banner.controlsAutoHide = true;
```

ScreenBannerRotator.controlsOrder

Usage

```
my_banner.controlsOrder
```

Description

Property; specifies which navigation buttons to show in the controls, and defines the order in which the buttons are displayed (separated by commas). This property can be any combination of the following values: "playpause", "previous", "next", "timer". The default value is undefined.

Example

The following example sets the controlsOrder property to "playpause,timer":

```
my_banner.controlsOrder = "playpause,timer";
```

ScreenBannerRotator.delayTime

Usage

```
my_banner.delayTime
```

Description

Property; amount of delay between images, in seconds. This property can be any positive number or 0. The default value is 5.

Example

The following example sets the delayTime property to 3:

```
my_banner.delayTime = 3;
```

ScreenBannerRotator.gotoImageAt()

Usage

```
my_banner.gotoImageAt(index)
```

Parameters

index An integer indicating the index of the image in the gallery. This is a zero-based index, so 0 retrieves the first image, 1 retrieves the second image, and so on.

Returns

Nothing.

Description

Method; navigates to the image with the specified index.

Example

The following example demonstrates how to navigate to the image with index 5:

```
my_banner.gotoImageAt(5);
```

ScreenBannerRotator.gotoNextImage()

Usage

```
my_banner.gotoNextImage()
```

Returns

Nothing.

Description

Method; navigates to the next image in the gallery.

Example

The following example demonstrates how to navigate to the next image in the gallery:

```
my_banner.gotoNextImage();
```

ScreenBannerRotator.gotoPreviousImage()

Usage

```
my_banner.gotoPreviousImage()
```

Returns

Nothing.

Description

Method; navigates to the previous image in the gallery.

Example

The following example demonstrates how to navigate to the previous image in the gallery:

```
my_banner.gotoPreviousImage();
```

ScreenBannerRotator.pauseOnMouseOver

Usage

```
my_banner.pauseOnMouseOver
```

Description

Property; specifies whether the banner should pause rotating images when the mouse is over its area ("true") or not ("false"). The default value is "false".

Example

The following example sets the `pauseOnMouseOver` property to "true":

```
my_banner.pauseOnMouseOver = true;
```

ScreenBannerRotator.randomImages

Usage

```
my_banner.randomImages
```

Description

Property; determines whether to display images in sequential ("false") or random ("true") order. The default value is "false".

Example

The following example sets the `randomImages` property to "true":

```
my_banner.randomImages = true;
```

ScreenBannerRotator.randomTransitions

Usage

```
my_banner.randomTransitions
```

Description

Property; determines whether to apply transition types (if multiple values are selected) in sequential ("false") or random ("true") order. The default value is "true".

Example

The following example sets the `randomTransitions` property to "false":

```
my_banner.randomTransitions = false;
```

ScreenBannerRotator.repeatMode

Usage

```
my_banner.repeatMode
```

Description

Property; defines the banner behavior when it reaches the last image. This property can be either "true" or "false". The default value is "true".

Example

The following example sets the `repeatMode` property to "false":

```
my_banner.repeatMode = false;
```

ScreenBannerRotator.showControls

Usage

```
my_banner.showControls
```

Description

Property; determines if the controls are visible. The default value is "true".

Example

The following example sets the showControls property to "false":

```
my_banner.showControls = false;
```

ScreenBannerRotator.showControlsScreen

Usage

```
my_banner.showControlsScreen
```

Description

Property; determines whether to show the background (screen) of the controls ("true") or not ("false"). The default value is "true".

Example

The following example sets the showControlsScreen property to "false":

```
my_banner.showControlsScreen = false;
```

ScreenBannerRotator.showDescription

Usage

```
my_banner.showDescription
```

Description

Property; determines if the description area is visible. The default value is "true".

Example

The following example sets the showDescription property to "false":

```
my_banner.showDescription = false;
```

ScreenBannerRotator.showNumberButtons

Usage

```
my_banner.showNumberButtons
```

Description

Property; determines whether to show the number buttons in the controls ("true") or not ("false"). The default value is "true".

Example

The following example sets the showNumberButtons property to "false":

```
my_banner.showNumberButtons = false;
```

ScreenBannerRotator.transitionType

Usage

```
my_banner.transitionType
```

Description

Property; specifies the type of transition to be applied when rotating images. Multiple values should be separated by commas, without blank space after commas. This property can be any combination of the following values: "fade", "blinds", "windows". The default value is "fade,blinds,windows".

Example

The following example sets the transitionType property to "fade,blinds":

```
my_banner.transitionType = "fade,blinds";
```

ScreenBannerRotator.xmlFileURL**Usage**

```
my_banner.xmlFileURL
```

Description

Property; the path to the XML file. The default value is "settings.xml".

Example

The following example sets the xmlFileURL property to "data/banner.xml":

```
my_banner.xmlFileURL = "data/banner.xml";
```

Hints and Tips

1. Changing the font for the description text.

The Screen Banner Rotator uses Arial as the default font for the description text. To change the default font, open the source FLA file (ScreenBannerRotator.fla) and do the following:

1. Open "Screen Banner Rotator" folder > "Description items" folder in the Library panel, and select "description text" symbol.
2. Select the dynamic text field in "description text" symbol, and choose the font you would like to use.
3. By default, the "Uppercase", "Lowercase", "Numerals" and "Punctuation" character sets for the dynamic text field are embedded. If you want to use an embedded font, leave the above character sets selected.

2. Using special characters such as ã¸¸ in the description text.

You need to embed these special characters in the text field first. To do this, open the source FLA file (ScreenBannerRotator.fla) and make the following steps:

1. Open "Screen Banner Rotator" folder > "Description items" folder in the Library panel, and select "description text" symbol.
2. Select the dynamic text field in "description text" symbol. In the properties panel, press the "Embed" button and you will then see a field labeled "Include these characters" — paste the special characters that you need here.

3. Formatting HTML text in XML (bold and italic styles).

You need to embed the characters for bold and italic styles first. To do this, open the source FLA file (ScreenBannerRotator.fla) and make the following steps:

1. Create a blank text field with the same font as the description text field. Leave the instance name blank.
2. With the text field selected, select the Bold icon in the properties panel. Now press the "Embed" button and select "Uppercase", "Lowercase", "Numerals" and "Punctuation" character sets and press "OK".
3. Create a new text field and select the Italic icon. Repeat the same embed steps.