About 3D ImageFlow Gallery

Dazzle your viewers with 3D photo navigation. Create an amazing gallery with cool perspective effects in seconds and give your photos stunning 3d and camera effects. The component is fully ActionScript 2 compatible.

Want to know how it works? Read the extensive **documentation** and complete ActionScript 2 object **reference**.

Introduction

This reference allows you to build Flash movies with **3D Image Flow Gallery** and your own components using ActionScript 2. On the next page you'll find a summary of the contents of this reference.

Gallery class

Inheritance MovieClip > UIObject > UIComponent > Gallery

ActionScript Class Name com.flzone.imageflow.Gallery

An *item* is an ActionScript object used for storing the units of information in the gallery. The gallery can be thought of as an array; each indexed space of the array is an item.
An item is an object that has an *url* property that is the path to the image to show, optionally an item can have a *width* and *height* property which contains the image size, a *description* property that is used to describe the image and is shown when the image is selected and a *link* and *target* property that contains the url where to go to when the selected image is double clicked.

To add a gallery component to the tab order of an application, set its `tabIndex` property (see UIComponent.tabIndex).

Method summary for the Gallery class

The following table lists methods of the Gallery class.

Method	Description
Gallery.addItem()	Adds an item to the end of the gallery.
Gallery.addItemAt()	Adds an item to the gallery at the specified index.
Gallery.flip()	Navigate to the next item depending on the last direction.
Gallery.getItemAt()	Return the item at the specified index.
Gallery.goto()	Navigate to the specified index.
Gallery.load()	Loads the xml specified by the xmlPath property.
Gallery.next()	Navigate to the next item.
Gallery.previous()	Navigate to the previous item.
Gallery.removeAll()	Removes all items from the gallery.
Gallery.removeItemAt()	Removes the item at the specified index.
Gallery.replaceItemAt()	Replaces the item at the specified index with another item.

Methods inherited from the UIObject class

The following table lists the methods the List class inherits from the UIObject class. When calling these methods, use the form *galleryInstance.methodName*.

Method	Description
UIObject.createClassObject()	Creates an object on the specified class.
UIObject.createObject()	Creates a subobject on an object.
UIObject.destroyObject()	Destroys a component instance.
UIObject.doLater()	Calls a function when parameters have been set in the Property and Component inspectors.
UIObject.getStyle()	Gets the style property from the style declaration or object.
UIObject.invalidate()	Marks the object so it is redrawn on the next frame interval.
UIObject.move()	Moves the object to the requested position.
UIObject.redraw()	Forces validation of the object so it is drawn in the current frame.

Methods inherited from the UIComponent class

The following table lists the methods the List class inherits from the UIComponent class. When calling these methods, use the form *galleryInstance.methodName*.

Method	Description
UIComponent.getFocus()	Returns a reference to the object that has focus.
UIComponent.setFocus()	Sets focus to the component instance.

Property summary for the Gallery class

The following table lists properties of the Gallery class.

Property	Description
Gallery.autoFlip	Automatically navigate to the next item.
Gallery.autoFlipInterval	The duration an item should is selected before navigating to the next item.
Gallery.backgroundColor	The background color of the gallery.
Gallery.backgroundTransparent	A boolean indicating if the background should be transparent.
Gallery.bytesLoaded	A read-only property that indicates the number of bytes that have been loaded.
Gallery.bytesTotal	A read-only property that indicates the total number of bytes in the xml.
Gallery.descriptionColor	A number indicating the color of the description text.
Gallery.descriptionFont	The font used for the description text.
Gallery.descriptionSize	The size of the description text.
Gallery.dofStrength	The strength of the depth of field effect.
Gallery.flipDuration	The duration in milliseconds the navigation animation lasts.
Gallery.fogStrength	The strength of the distance fog.
Gallery.imageAngle	The angle of the items in the back.
Gallery.imageDepth	The depth for the items in the back.
Gallery.imageHeight	The default height for an item.
Gallery.imageKeepAspect	A boolean indicating whether to keep aspect ratio in the items, changes the default height.
Gallery.imageWidth	The default width for an item.
Gallery.imageOffset	A number indicating the offset from the center of the gallery to the bottom of the item.
Gallery.imageQuality	The quality of the gallery items.
Gallery.imageSpaceBack	The space between the items in the back.
Gallery.imageSpaceMain	The space between the selected item and the images in the back.
Gallery.items	An array with the gallery items. This property is write-only.
Gallery.length	The number of items in the gallery. This property is read-only.
Gallery.motionBlur	A boolean value indicating whether to use motionblur when navigating.
Gallery.percentLoaded	A number that indicates the percentage of the xml is loaded. This property is read-only.
Gallery.reflections	A boolean indicating whether the items have reflection.
Gallery.reflectionSize	The size of the reflection.
Gallery.reflectionStrength	The strength of the reflection.
Gallery.selectedIndex	The index of a selection in the gallery.
Gallery.selectedItem	The selected item in the gallery. This property is read-only.
Gallery.titleColor	A number indicating the color of the title text.
Gallery.titleFont	The font used for the title text.
Gallery.titleSize	The size of the title text.

Gallery.titleText	The text in the title.
Gallery.xmlPath	A string that indicates the URL of the xml to be loaded.

Properties inherited from the UIObject class

The following table lists the methods the List class inherits from the UIObject class. When calling these methods, use the form *galleryInstance.methodName*.

Property	Description
UIObject.bottom	The position of the bottom edge of the object, relative to the bottom edge of its parent. Read-only.
UIObject.height	The height of the object, in pixels. Read-only.
UIObject.left	The left edge of the object, in pixels. Read-only.
UIObject.right	The position of the right edge of the object, relative to the right edge of its parent. Read-only.
UIObject.scaleX	A number indicating the scaling factor in the *x* direction of the object, relative to its parent.
UIObject.scaleY	A number indicating the scaling factor in the *y* direction of the object, relative to its parent.
UIObject.top	The position of the top edge of the object, relative to its parent. Read-only.
UIObject.visible	A Boolean value indicating whether the object is visible (`true`) or not (`false`).
UIObject.width	The width of the object, in pixels. Read-only.
UIObject.x	The left edge of the object, in pixels. Read-only.
UIObject.y	The top edge of the object, in pixels. Read-only.

Properties inherited from the UIComponent class

The following table lists the properties the List class inherits from the UIComponent class. When accessing these properties, use the form *galleryInstance.propertyName*.

Property	Description
UIComponent.enabled	Indicates whether the component can receive focus and input.
UIComponent.tabIndex	A number indicating the tab order for a component in a document.

Event summary for the Gallery class

The following table lists events that of the Gallery class.

Event	Description
Gallery.change	Broadcast whenever user interaction causes the selection to change.
Gallery.click	Broadcast when front image is clicked.
Gallery.complete	Triggered when the content finished loading.
Gallery.doubleClick	Broadcast when front image is double clicked.
Gallery.motionFinished	Triggered when the tween animation finished.
Gallery.progress	Triggered while content is loading.

Events inherited from the UIObject class

The following table lists the events the List class inherits from the UIObject class.

Event	Description
UIObject.draw	Broadcast when an object is about to draw its graphics.
UIObject.hide	Broadcast when an object's state changes from visible to invisible.
UIObject.load	Broadcast when subobjects are being created.
UIObject.move	Broadcast when the object has moved.
UIObject.resize	Broadcast when an object has been resized.
UIObject.reveal	Broadcast when an object's state changes from invisible to visible.

UIObject.unload	Broadcast when the subobjects are being unloaded.

Events inherited from the UIComponent class

The following table lists the events the List class inherits from the UIComponent class.

Event	Description
UIComponent.focusIn	Broadcast when an object receives focus.
UIComponent.focusOut	Broadcast when an object loses focus.
UIComponent.keyDown	Broadcast when a key is pressed.
UIComponent.keyUp	Broadcast when a key is released.

Gallery.addItem()

Usage

```
galleryInstance.addItem(itemObject)
```

Parameters

itemObject An item object that has an `url` property that is the path to the image to show, optionally an item can have a `width` and `height` property which contains the image size, a `description` property that is used to describe the image and is shown when the image is selected and a `link` and `target` property that contains the url where to go to when the selected image is double clicked.

Returns

Nothing.

Description

Method; adds a new item to the end of the gallery.

Example

The following code will add an item to the `my_gallery` instance. To try this code, drag a List component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:”me.jpg”, description:”A photo of me”});
```

Gallery.addItemAt()

Usage

```
galleryInstance.addItemAt(index, itemObject)
```

Parameters

Index A number greater than or equal to 0 that indicates the position of the item.

itemObject An item object that has an `url` property that is the path to the image to show, optionally an item can have a `width` and `height` property which contains the image size, a `description` property that is used to describe the image and is shown when the image is selected and a `link` and `target` property that contains the url where to go to when the selected image is double clicked.

Returns

Nothing.

Description

Method; adds a new item to the position specified by the `index` parameter.

Example

The following example adds an item to the first index position, which is the second item in the gallery and will place photo3.jpg between photo1.jpg and photo2.jpg. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:”photo1.jpg”});
my_gallery.addItem({url:”photo2.jpg”});
my_gallery.addItemAt(1, {url:”photo3.jpg”});
```

Gallery.flip()

Usage

```
galleryInstance.flip()
```

Returns

Nothing.

Description

Method; Navigate to the next item depending on the last direction. Direction will chance when the end or begin of the gallery is reached.

Example

The following example lets you navigate through the gallery using a button. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Next, drag a Button component to the Stage and give it the instance name **flip_button**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;
var flip_button:mx.controls.Button;

flip_button.label = “flip”;

my_gallery.addItem({url:”photo1.jpg”});
my_gallery.addItem({url:”photo2.jpg”});
my_gallery.addItem({url:”photo3.jpg”});

var buttonListener:Object = new Object();
buttonListener.click = function(evt_obj:Object) {
    my_gallery.flip();
};
flip_button.addEventListener(“click”, buttonListener);
```

Gallery.getItemAt()

Usage

```
galleryInstance.getItemAt(index)
```

Parameters

Index A number greater than or equal to 0, and less than Gallery.length. It specifies the index of the item to retrieve.

Returns

The indexed item object; undefined if the index is out of range.

Description

Method; retrieves the item at the specified index.

Example

The following code displays the url of the item at index position 2. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:”photo1.jpg”});
my_gallery.addItem({url:”photo2.jpg”});
my_gallery.addItem({url:”photo3.jpg”});

trace(my_gallery.getItemAt(2).url);
```

Gallery.goto()

Usage

```
galleryInstance.goto(index)
```

Parameters

Index A number greater than or equal to 0, and less than Gallery.length. It specifies the index of the item to navigate to.

Returns

Nothing.

Description

Method; Navigate to the specified index in the gallery.

Example

The following code navigates the gallery using a NumericStepper. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Next, drag a NummericStepper component to the Stage and give it the instance name **my_nstep**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;
var my_nstep:mx.controls.NummericStepper;

my_gallery.addItem({url:”photo1.jpg”});
my_gallery.addItem({url:”photo2.jpg”});
my_gallery.addItem({url:”photo3.jpg”});

var nstepListener:Object = new Object();
nstepListener.change = function(evt_obj:Object) {
    my_gallery.goto(evt_obj.target.value);
};
my_nstep.addEventListener(“change”, nstepListener);
```

Gallery.load()

Usage

```
galleryInstance.load([xmlPath])
```

Parameters

xmlPath An optional parameter that specifies the value of the xmlPath property before the load begins. If a value is not specified, the current value of xmlPath is used as is.

Returns

Nothing.

Description

Method; Tells the gallery to begin loading the xml.

Example

The following example creates a Gallery instance, my_gallery, and a Button instance and sets the c. Add the xmlPath to the location where the gallery xml is located. Next the example creates a listener for the click event on the button. When the user clicks the button, the event handler calls the my_gallery.load() to load the xml.

Drag a Gallery component and a Button component from the Component panel to the Library, then add the following code to Frame 1 in the timeline:

```
this.createClassObject(com.flzone.imageflow.Gallery, " my_gallery ", 10);
this.createClassObject(mx.controls.Button, "load_button", 20, {label:"Load"});

my_gallery.xmlPath = "gallery.xml";

var buttonListener:Object = new Object();
buttonListener. click = function(evt_obj:Object) {
    my_gallery.load();
};
load_button.addEventListener("click", buttonListener);
```

Gallery.next()

Usage

```
galleryInstance.next([step])
```

Parameters

step A optional parameter that specifies the amount to change. The default value is 1. Will move forward through the gallery.

Returns

Nothing.

Description

Method; Navigate to the next item in the gallery.

Example

The following example lets you move forward through the gallery using a button. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Next, drag a Button component to the Stage and give it the instance name **next_button**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;
var next_button:mx.controls.Button;

prev_button.label = “Next”;

my_gallery.addItem({url:”photo1.jpg”});
my_gallery.addItem({url:”photo2.jpg”});
my_gallery.addItem({url:”photo3.jpg”});

var buttonListener:Object = new Object();
buttonListener.click = function(evt_obj:Object) {
    my_gallery.next();
};
next_button.addEventListener(“click”, buttonListener);
```

Gallery.previous()

Usage

```
galleryInstance.previous([step])
```

Parameters

step A optional parameter that specifies the amount to change. The default value is 1. Will move backwards through the gallery.

Returns

Nothing.

Description

Method; Navigate to the previous item in the gallery.

Example

The following example lets you move backwards through the gallery using a button. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Next, drag a Button component to the Stage and give it the instance name **previous_button**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;
var previous_button:mx.controls.Button;

previous_button.label = “Previous”;

my_gallery.addItem({url:”photo1.jpg”});
my_gallery.addItem({url:”photo2.jpg”});
my_gallery.addItem({url:”photo3.jpg”});

var buttonListener:Object = new Object();
buttonListener.click = function(evt_obj:Object) {
    my_gallery.next();
};
previous_button.addEventListener(“click”, buttonListener);
```

Gallery.removeAll()

Usage

```
galleryInstance.removeAll()
```

Returns

Nothing.

Description

Method; Removes all items in the gallery.

Example

The following code clears all items in the gallery when a button is clicked. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Next, drag a Button component to the Stage and give it the instance name **remove_button**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;
var remove_button:mx.controls.Button;

remove_button.label = “Remove”;

my_gallery.addItem({url:”photo1.jpg”});
my_gallery.addItem({url:”photo2.jpg”});
my_gallery.addItem({url:”photo3.jpg”});

var buttonListener:Object = new Object();
buttonListener.click = function(evt_obj:Object) {
    my_gallery.removeAll();
    evt_obj.target.enabled = false;
};
remove_button.addEventListener(“click”, buttonListener);
```

Gallery.removeItemAt()

Usage

```
galleryInstance.removeItemAt(index)
```

Parameters

Index A number greater than or equal to 0 and less than Gallery.length that indicates the position of the item to remove.

Returns

Nothing.

Description

Method; Removes the item at the specified index position. The gallery indices after the specified index collapse by one.

Example

The following code removes the selected item in a Gallery component when a button is clicked. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Next, drag a Button component to the Stage and give it the instance name **remove_button**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;
var remove_button:mx.controls.Button;

remove_button.label = “Remove”;

my_gallery.addItem({url:”photo1.jpg”});
my_gallery.addItem({url:”photo2.jpg”});
my_gallery.addItem({url:”photo3.jpg”});

var buttonListener:Object = new Object();
buttonListener.click = function(evt_obj:Object) {
    my_gallery.removeItemAt(my_gallery.selectedIndex);
};
remove_button.addEventListener(“click”, buttonListener);
```

Gallery.replaceItemAt()

Usage

```
galleryInstance.replaceItemAt(index, itemObject)
```

Parameters

Index A number greater than or equal to 0 and less than Gallery.length that indicates the position of the item to place.

itemObject An item object that has an `url` property that is the path to the image to show, optionally an item can have a `width` and `height` property which contains the image size, a `description` property that is used to describe the image and is shown when the image is selected and a `link` and `target` property that contains the url where to go to when the selected image is double clicked.

Returns

Nothing.

Description

Method; Removes the item at the specified index position. The gallery indices after the specified index collapse by one.

Example

The following example replaces the current selected item in the Gallery. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Next, drag a Button component to the Stage and give it the instance name **replace_button**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;
var replace_button:mx.controls.Button;

replace_button.label = “Replace”;

my_gallery.addItem({url:”photo1.jpg”});
my_gallery.addItem({url:”photo2.jpg”});
my_gallery.addItem({url:”photo3.jpg”});

var buttonListener:Object = new Object();
buttonListener.click = function(evt_obj:Object) {
    my_gallery.replaceItemAt(my_gallery.selectedIndex, {url:”photo4.jpg”});
};
replace_button.addEventListener(“click”, buttonListener);
```

Gallery.autoFlip

Usage

```
galleryInstance.autoFlip
```

Description

Property; a Boolean value that indicates if the gallery should automatically with an interval navigate to the next item. The default value is **false**.

Example

The following example will let the gallery automatically flip to the next image with an interval of 2000 milliseconds. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:”photo1.jpg”});
my_gallery.addItem({url:”photo2.jpg”});
my_gallery.addItem({url:”photo3.jpg”});

my_gallery.autoFlipInterval = 2000;
my_gallery.autoFlip = true;
```

Gallery.autoFlipInterval

Usage

```
galleryInstance.autoFlipInterval
```

Description

Property; the time in milliseconds an item will be selected before continuing to the next when the Gallery.autoFlip property is set to true. The default value is **3000**.

Example

The following example will let the gallery automatically flip to the next image with an interval of 2000 milliseconds. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:”photo1.jpg”});
my_gallery.addItem({url:”photo2.jpg”});
my_gallery.addItem({url:”photo3.jpg”});

my_gallery.autoFlipInterval = 2000;
my_gallery.autoFlip = true;
```

Gallery.backgroundColor

Usage

```
galleryInstance.backgroundColor
```

Description

Property; a number that represents the RGB numeric value for the background of the gallery. This property will have no effect when Gallery.backgroundTransparent is set to true. The default value is **0x000000**(black).

Example

The following example will make the background of the gallery white. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.backgroundColor = 0xFFFFFF;

my_gallery.addItem({url:”photo1.jpg”});
my_gallery.addItem({url:”photo2.jpg”});
my_gallery.addItem({url:”photo3.jpg”});
```

Gallery.backgroundTransparent

Usage

```
galleryInstance.backgroundTransparent
```

Description

Property; a Boolean value indicating if the background of the gallery will be transparent. When this property is set then the Gallery.backgroundColor property is ignored. Also the transparency has influence on the reflections and distance fog. The default value is **false**.

Example

The following example will make the background of the gallery transparent. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.backgroundTransparent = true;

my_gallery.addItem({url:"photo1.jpg"});
my_gallery.addItem({url:"photo2.jpg"});
my_gallery.addItem({url:"photo3.jpg"});
```

Gallery.bytesLoaded

Usage

```
galleryInstance.bytesLoaded
```

Description

Property (read-only); the number of bytes the xml is loaded. The default value is **0** until xml begins loading.

Example

With a Gallery component and a ProgressBar component in the Library of the current document, the following code creates a progress bar an gallery instances. It then creates a listener object with a progress event handler that shows the progress of the load. The listener is registered to the my_gallery instance. Add the following code to Frame 1 in the timeline:

```
import com.flzone.imageflow.Gallery;
import mx.controls.ProgressBar;

this.createClassObject(Gallery, “my_gallery”, 10);
this.createClassObject(ProgressBar, “my_pb”, 20, {source:”my_gallery”});

my_gallery.move(1, 50);
my_pb.move(1, 1);

var loaderListener:Object = new Object();
loaderListener.progress = function(evt_obj:Object) {
    my_pb.setProgress(my_gallery.bytesLoaded, my_gallery.bytesTotal);
};
my_gallery.addEventListener(“progress”, loaderListener);
my_gallery.load(“gallery.xml”);
```

Gallery.bytesTotal

Usage

```
galleryInstance.bytesTotal
```

Description

Property (read-only); the size of the xml, in bytes. The default value is **0** until xml begins loading.

Example

With a Gallery component and a ProgressBar component in the Library of the current document, the following code creates a progress bar an gallery instances. It then creates a listener object with a progress event handler that shows the progress of the load. The listener is registered to the my_gallery instance. Add the following code to Frame 1 in the timeline:

```
import com.flzone.imageflow.Gallery;
import mx.controls.ProgressBar;

this.createClassObject(Gallery, “my_gallery”, 10);
this.createClassObject(ProgressBar, “my_pb”, 20, {source:”my_gallery”});

my_gallery.move(1, 50);
my_pb.move(1, 1);

var loaderListener:Object = new Object();
loaderListener.progress = function(evt_obj:Object) {
    my_pb.setProgress(my_gallery.bytesLoaded, my_gallery.bytesTotal);
};
my_gallery.addEventListener(“progress”, loaderListener);
my_gallery.load(“gallery.xml”);
```

Gallery.descriptionColor

Usage

```
galleryInstance.descriptionColor
```

Description

Property; a number that represents the RGB numeric value for the description text. The default value is **0xFFFFFF**(white).

Example

The following example will make the description text red. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.descriptionColor = 0xFF0000;
```

Gallery.descriptionFont

Usage

```
galleryInstance.descriptionFont
```

Description

Property; the font used for the description text. The default value is **Verdana**.

Example

The following example will set the font of the description text to Arial. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.descriptionFont = "Arial";
```

Gallery.descriptionSize

Usage

```
galleryInstance.descriptionSize
```

Description

Property; the font size used for the description text. The default value is **14**.

Example

The following example will set the font size of the description text to 12. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.descriptionSize = 12;
```

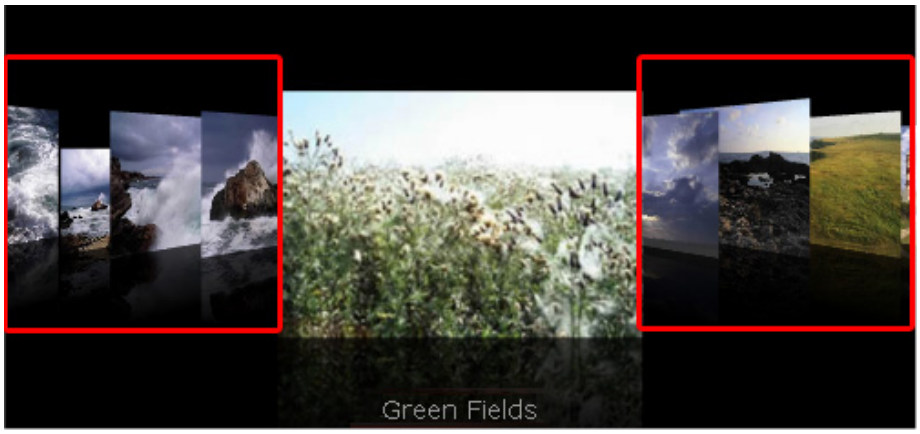
Gallery.dofStrength

Usage

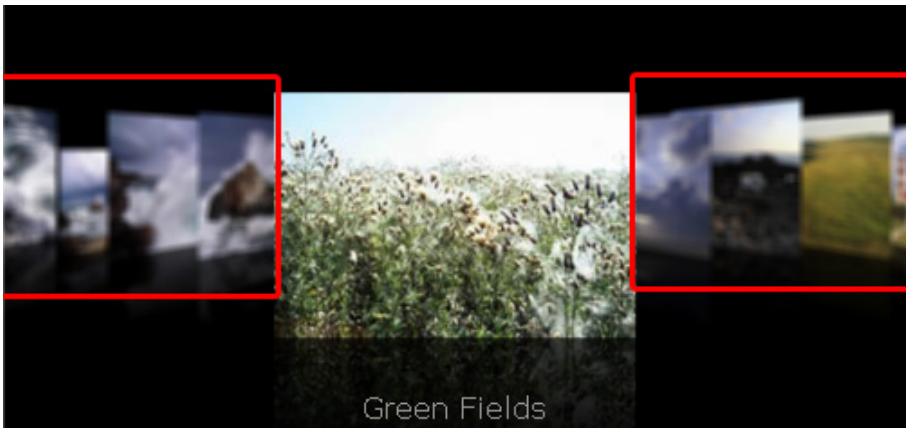
```
galleryInstance.dofStrength
```

Description

Property; the strength of the depth of field effect. The depth of field effect will make the items more blurry when they are further away. The default value is **0**, which leads to the following result:



dofStrength set to **5**:



Example

The following example will set the depth of field strength to 8. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.dofStrength = 8;
```

Gallery.flipDuration

Usage

```
galleryInstance.flipDuration
```

Description

Property; the duration in milliseconds the navigation animation lasts. The default value is **2000**.

Example

The following example lets you navigate through the gallery using a button, the speed of the navigation animation is set to 500 milliseconds. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Next, drag a Button component to the Stage and give it the instance name **flip_button**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;
var flip_button:mx.controls.Button;

flip_button.label = “flip”;

my_gallery.addItem({url:”photo1.jpg”});
my_gallery.addItem({url:”photo2.jpg”});
my_gallery.addItem({url:”photo3.jpg”});

my_gallery.flipDuration = 500;

var buttonListener:Object = new Object();
buttonListener.click = function(evt_obj:Object) {
    my_gallery.flip();
};
flip_button.addEventListener(“click”, buttonListener);
```

Gallery.fogStrength

Usage

```
galleryInstance.fogStrength
```

Description

Property; the strength of the distance fog. The distance fog will make the items slowly disappear when they are further away. The default value is **0**.

fogStrength set to **60**:

Example

The following example will set the distance fog strength to 50. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.fogStrength = 50;
```

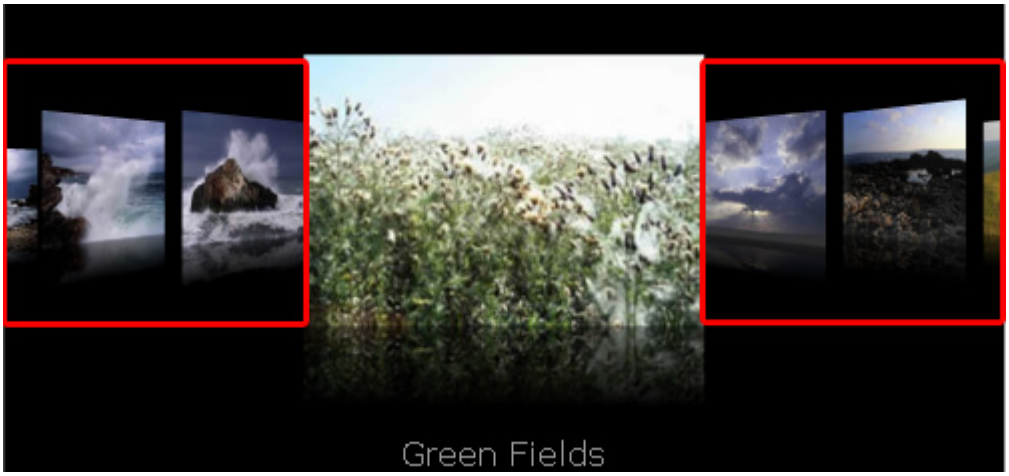
Gallery.imageAngle

Usage

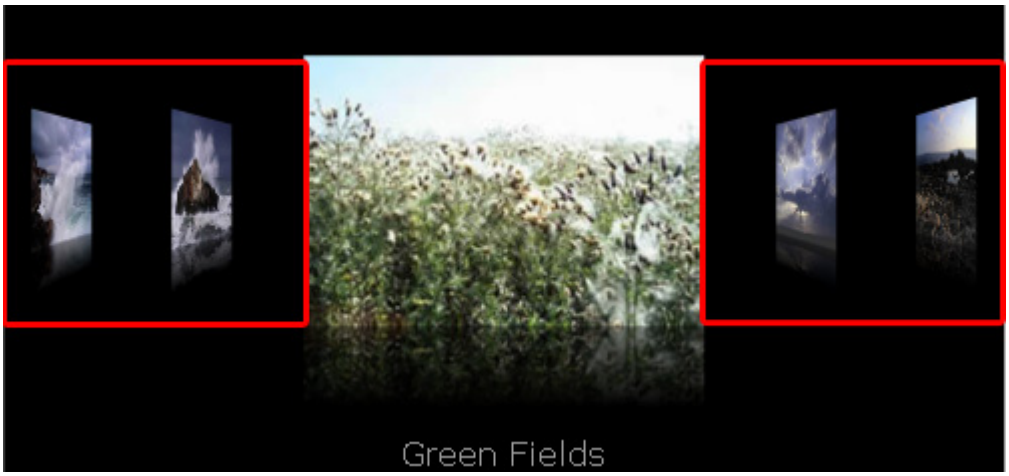
```
galleryInstance.imageAngle
```

Description

Property; The angle of the items in the back. The default value is **45:**



Gallery.imageAngle set to **70**:



Example

To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.imageAngle = 85;
```

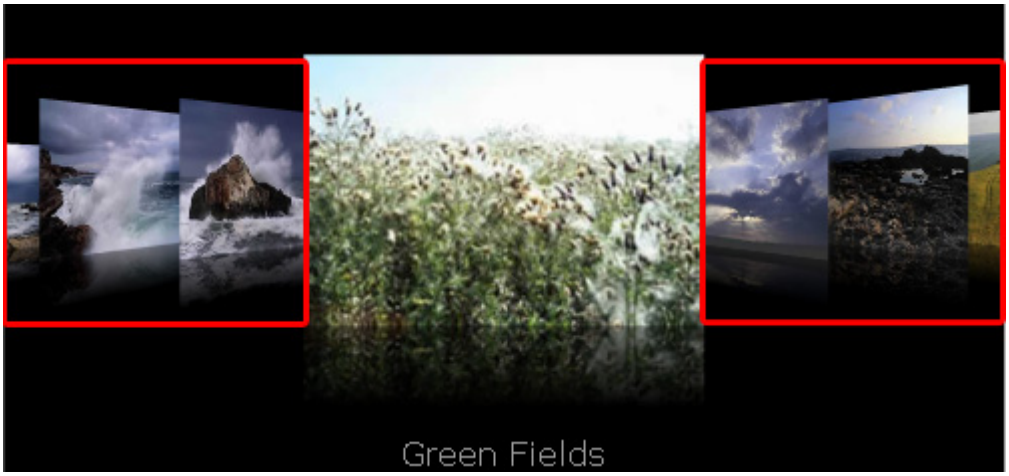
Gallery.imageDepth

Usage

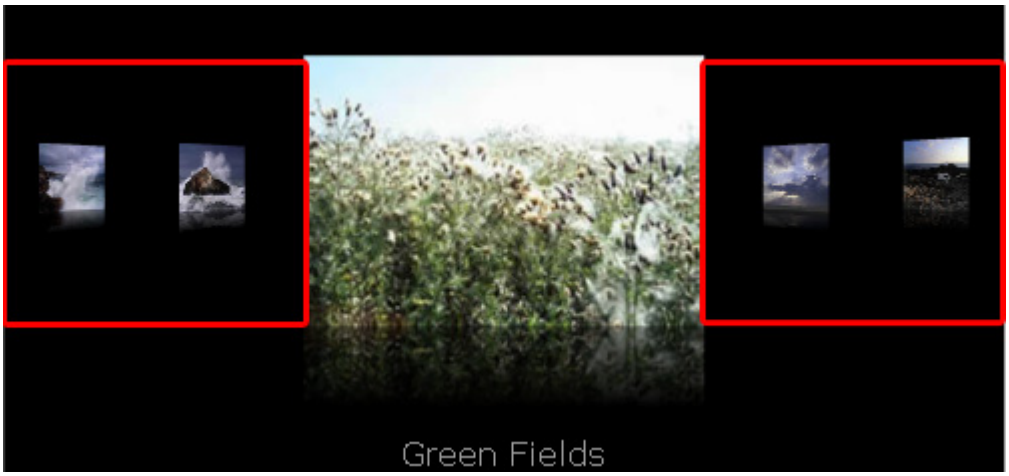
```
galleryInstance.imageDepth
```

Description

Property; The depth for the items in the back. The default value is **300**.



Gallery.imageDepth set to **1000;**



Example

To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.imageDepth = 1000;
```

Gallery.imageHeight

Usage

```
galleryInstance.imageHeight
```

Description

Property; The default height for an item. The default value is **200**. The imageHeight is overridden when the height on an individual item is set or when Gallery.imageKeepAspect is set to true.

Example

To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.setSize(500, 200);

my_galelry.imageOffset = 75;
my_gallery.imageWidth = 150;
my_gallery.imageHeight = 150;
my_gallery.imageKeepAspect = false;

my_gallery.addItem({url:”me.jpg”, description:”This is me”});
my_gallery.addItem({url:”colleagues.jpg”, description:”My colleagues”});
my_gallery.addItem({url:”office.jpg”, description:”The office”});
```

Gallery.imageKeepAspect

Usage

```
galleryInstance.imageKeepAspect
```

Description

Property; A boolean indicating whether to keep aspect ratio in the items, changes the height of an item. The default value is **true**.

Example

To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.setSize(500, 200);

my_galelry.imageOffset = 75;
my_gallery.imageWidth = 150;
my_gallery.imageHeight = 150;
my_gallery.imageKeepAspect = false;

my_gallery.addItem({url:”me.jpg”, description:”This is me”});
my_gallery.addItem({url:”colleagues.jpg”, description:”My colleagues”});
my_gallery.addItem({url:”office.jpg”, description:”The office”});
```

Gallery.imageWidth

Usage

```
galleryInstance.imageWidth
```

Description

Property; The default width for an item. The default value is **200**.

Example

To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.setSize(500, 200);

my_galelry.imageOffset = 75;
my_gallery.imageWidth = 150;
my_gallery.imageHeight = 150;
my_gallery.imageKeepAspect = false;

my_gallery.addItem({url:”me.jpg”, description:”This is me”});
my_gallery.addItem({url:”colleagues.jpg”, description:”My colleagues”});
my_gallery.addItem({url:”office.jpg”, description:”The office”});
```

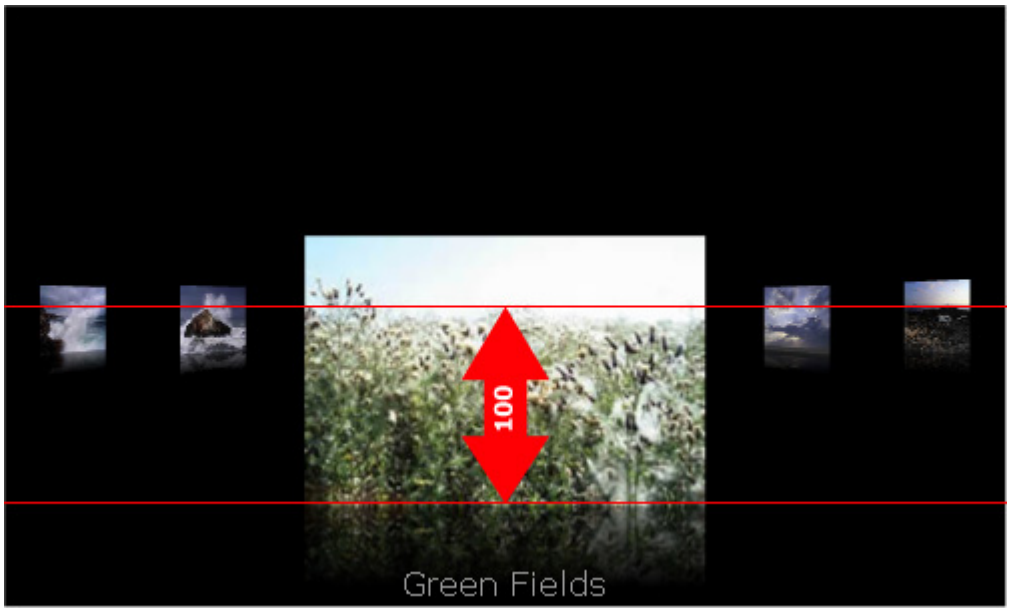
Gallery.imageOffset

Usage

```
galleryInstance.imageOffset
```

Description

Property; A number indicating the offset from the center of the gallery to the bottom of the item. The default value is **100**.



Example

To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.setSize(500, 200);

my_galelry.imageOffset = 75;
my_gallery.imageWidth = 150;
my_gallery.imageHeight = 150;
my_gallery.imageKeepAspect = false;

my_gallery.addItem({url:”me.jpg”, description:”This is me”});
my_gallery.addItem({url:”colleagues.jpg”, description:”My colleagues”});
my_gallery.addItem({url:”office.jpg”, description:”The office”});
```

Gallery.imageQuality

Usage

```
galleryInstance.imageQuality
```

Description

Property; The quality of the gallery items. Allowed values are “low”, “medium”, “high” and “best”. The default value is **medium**.

Example

To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

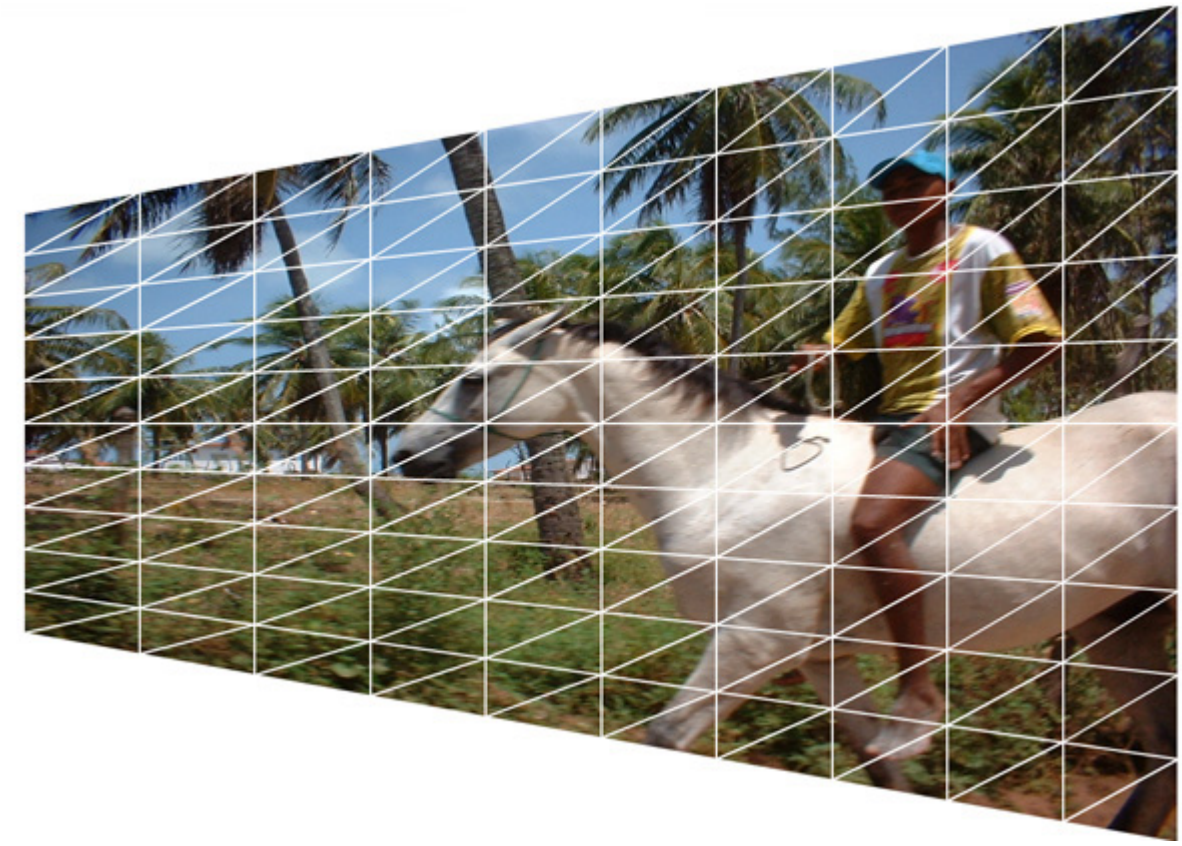
```
var my_gallery:com.flzone.imageflow.Gallery;

my_galelry.imageQuality = “high”;

my_gallery.addItem({url:”me.jpg”, description:”This is me”});
my_gallery.addItem({url:”colleagues.jpg”, description:”My colleagues”});
my_gallery.addItem({url:”office.jpg”, description:”The office”});
```

The property **imageQuality** sets the quality of the images on the background. Quality set to **low** results in a deformed image but has very good performance.

Quality set to **high** has a very good image representation but a slow performance.



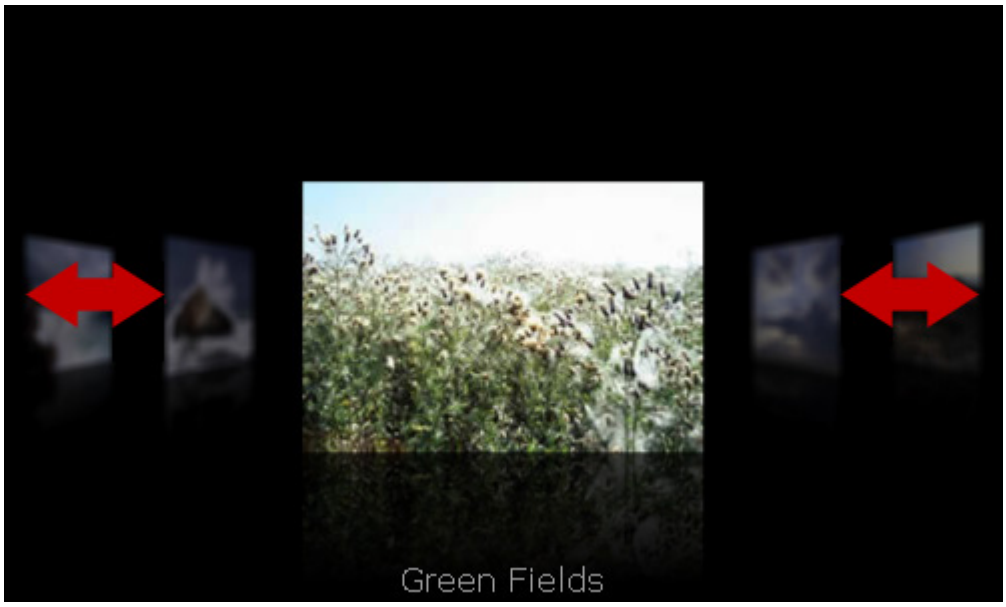
Gallery.imageSpaceBack

Usage

```
galleryInstance.imageSpaceBack
```

Description

Property; The space between the items in the back. The default value is **50**.



Example

To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_galelry.imageSpaceBack = 20;
my_galelry.imageSpaceMain = 100;

my_gallery.addItem({url:”me.jpg”, description:”This is me”});
my_gallery.addItem({url:”colleagues.jpg”, description:”My colleagues”});
my_gallery.addItem({url:”office.jpg”, description:”The office”});
```

Gallery.imageSpaceMain

Usage

```
galleryInstance.imageSpaceMain
```

Description

Property; The space between the selected item and the images in the back. The default value is **50**.

Example

To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_galelry.imageSpaceBack = 20;
my_galelry.imageSpaceMain = 100;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});
```

Gallery.items

Usage

```
galleryInstance.items
```

Description

Property (write-only); Accepts an array with the gallery items.

Example

To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

var images:Array = new Array();

images.push({url:"me.jpg", description:"This is me"});
images.push({url:"colleagues.jpg", description:"My colleagues"});
images.push({url:"office.jpg", description:"The office"});

my_gallery.items = images;
```

Gallery.length

Usage

```
galleryInstance.length
```

Description

Property (read-only); the number of items in the gallery.

Example

To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:”me.jpg”, description:”This is me”});
my_gallery.addItem({url:”colleagues.jpg”, description:”My colleagues”});
my_gallery.addItem({url:”office.jpg”, description:”The office”});

trace(“Number of items in the Gallery: “ + my_gallery.length);
```

Gallery.motionBlur

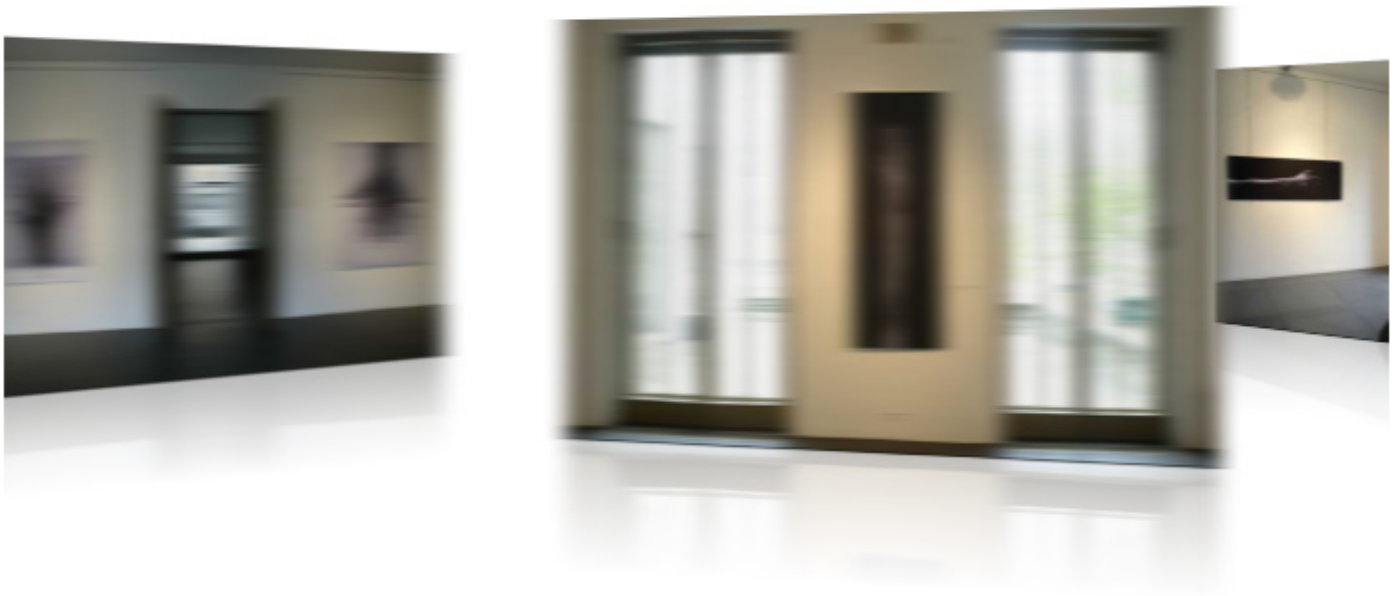
Usage

```
galleryInstance.motionBlur
```

Description

Property; a boolean value indicating whether to use motionblur when navigating. Default value is **false**.

The **motionBlur** property results in the following effect:



Without Motion Blur the transition would look like this:

Example

The following example turns on motion blur. To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.motionBlur = true;
```

Gallery.percentLoaded

Usage

```
galleryInstance.percentLoaded
```

Description

Property (read-only); a number that indicates the percentage of the xml is loaded. Typically, this property is used to present the progress to the user in an easily readable form. Use the following code to round the figure to the nearest integer:

```
Math.round(bytesLoaded/bytesTotal*100))
```

Example

The following example creates a listener object with a progress handler that traces the percent loaded and sends it to the output panel. To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

var loadListener:Object = new Object();
loadListener.progress = function(evt_obj:Object) {
    trace("The xml is " + my_gallery.percentLoaded + "% loaded.");
};
my_gallery.addEventListener("progress", loadListener);
my_gallery.load("gallery.xml");
```

Gallery.reflections

Usage

```
galleryInstance.reflections
```

Description

Property; a boolean indicating whether the items have reflection. The default value is **true**.

Example

The following example turns the reflections off. To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.reflections = false;
```

Gallery.reflectionSize

Usage

```
galleryInstance.reflectionSize
```

Description

Property; the size of the reflection, a number higher than 0 and lower or equal to 100, it is a percentage of the image height. The default value is **50**.

Reflection set to **30.**

Example

The following example turns the reflections on and sets the size and strength. To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.reflections = true;
my_gallery.reflectionSize = 30;
my_gallery.reflectionStrength = 80;
```

Gallery.reflectionStrength

Usage

```
galleryInstance.reflectionStrength
```

Description

Property; the size of the reflection, a number higher than 0 and lower or equal to 100, it is the alpha where the reflection start with. The default value is **30**.

Gallery.reflectionStrength set to **70:**

Example

The following example turns the reflections on and sets the size and strength. To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.reflections = true;
my_gallery.reflectionSize = 30;
my_gallery.reflectionStrength = 80;
```

Gallery.selectedIndex

Usage

```
galleryInstance.selectedIndex
```

Description

Property; the selected index of the gallery. If you assign a value to selectedIndex, the indicated item is selected.

Using the selectedIndex property to change selection doesn't dispatch a change event and will not animate to the item but will change directly. To dispatch the change event, use the following code:

```
my_gallery.dispatchEvent({type:"change", target:my_gallery});
```

Example

The following example selects the second item in the gallery and displays the index of the currently selected whenever the user navigates. To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.selectedIndex = 1;

var galleryListener:Object = new Object();
galleryListener.change = function(evt_obj:Object) {
    trace("selectedIndex = " + evt_obj.target.selectedIndex);
};
My_gallery.addEventListener("change", galleryListener);
```

Gallery.selectedItem

Usage

```
galleryInstance.selectedItem
```

Description

Property (read-only); the selected item in the gallery.

Example

The following example displays the description of the selected item:

```
trace(my_gallery.selectedItem.description);
```

The following example displays all properties of the selected item whenever the user navigates. To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

var galleryListener:Object = new Object();
galleryListener.change = function(evt_obj:Object) {
    var tempStr:String = "[object";
    for (var prop:String in evt_obj.target.selectedItem) {
        tempStr += " " + prop + ":'" + evt_obj.target.selectedItem[prop] + "'";
    }
    tempStr += "]";
    trace(tempStr);
};
My_gallery.addEventListener("change", galleryListener);
```

Gallery.titleColor

Usage

```
galleryInstance.titleColor
```

Description

Property; a number that represents the RGB numeric value for the title text. The default value is **0xFFFFFF**(white).

Example

The following example will set the title and change it appearance. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.titleText = "My Personal Gallery";
my_gallery.titleColor = 0xFF0000;
my_gallery.titleFont = "Arial";
my_gallery.titleSize = 36;
```

Gallery.titleFont

Usage

```
galleryInstance.titleColor
```

Description

Property; the font used for the title text. The default value is **Verdana**.

Example

The following example will set the title and change it appearance. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.titleText = "My Personal Gallery";
my_gallery.titleColor = 0xFF0000;
my_gallery.titleFont = "Arial";
my_gallery.titleSize = 36;
```

Gallery.titleSize

Usage

```
galleryInstance.titleSize
```

Description

Property; the text size of the title. The default value is **18**.

Example

The following example will set the title and change it appearance. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.titleText = "My Personal Gallery";
my_gallery.titleColor = 0xFF0000;
my_gallery.titleFont = "Arial";
my_gallery.titleSize = 36;
```

Gallery.titleText

Usage

```
galleryInstance.titleText
```

Description

Property; the text in the title. The default value is an empty string.

Example

The following example will set the title and change it appearance. To try this code, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

my_gallery.titleText = "My Personal Gallery";
my_gallery.titleColor = 0xFF0000;
my_gallery.titleFont = "Arial";
my_gallery.titleSize = 36;
```

Gallery.xmlPath

Usage

```
galleryInstance.xmlPath
```

Description

Property; a string that indicates the URL of the xml to be loaded.

Example

The following example creates a Gallery instance, my_gallery, and a Button instance and sets the c. Add the xmlPath to the location where the gallery xml is located. Next the example creates a listener for the click event on the button. When the user clicks the button, the event handler calls the my_gallery.load() to load the xml.

Drag a Gallery component and a Button component from the Component panel to the Library, then add the following code to Frame 1 in the timeline:

```
this.createClassObject(com.flzone.imageflow.Gallery, " my_gallery ", 10);
this.createClassObject(mx.controls.Button, "load_button", 20, {label:"Load"});

my_gallery.xmlPath = "gallery.xml";

var buttonListener:Object = new Object();
buttonListener.click = function(evt_obj:Object) {
    my_gallery.load();
};
load_button.addEventListener("click", buttonListener);
```

Gallery.change

Usage

Usage 1:

```
var listenerObject:Object = new Object();
listenerObject.change = function(eventObject:Object) {
    // Your code here
};
```

Usage 2:

```
on (change) {
    // Your code here
}
```

Description

Event; broadcast to all registered listeners when the selected index of the gallery changes as a result of user interaction.

The first usage example uses a dispatcher/listener event model. A component instance dispatches an event (in this case, change) and the event is handled by a function, also called a handler, on a listener object (listenerObject) that you create. You define a method with the same name as the event on the listener object; the method is called when the event is triggered. When the event is triggered, it automatically passes an event object (eventObject) to the listener object method. Each event object has properties that contain information about the event. You can use these properties to write code that handles the event. For more information, see EventDispatcher class (API).

Finally, you call the addEventListener() method on the component instance that broadcasts the event to register the listener with the instance. When the instance dispatches the event, the listener is called.

The second usage example uses an on() handler and must be attached directly to a Gallery instance. The keyword this, used inside an on() handler attached to a component, refers to the component instance.

Example

The following example displays the description of the selected item whenever the selection changes. To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

var galleryListener:Object = new Object();
galleryListener.change = function(evt_obj:Object) {
    trace(evt_obj.target.selectedItem.description);
};
My_gallery.addEventListener("change", galleryListener);
```

Gallery.click

Usage

Usage 1:

```
var listenerObject:Object = new Object();
listenerObject.click = function(eventObject:Object) {
    // Your code here
};
```

Usage 2:

```
on (click) {
    // Your code here
}
```

Description

Event; broadcast when front image is clicked.

The first usage example uses a dispatcher/listener event model. A component instance dispatches an event (in this case, click) and the event is handled by a function, also called a handler, on a listener object (listenerObject) that you create. You define a method with the same name as the event on the listener object; the method is called when the event is triggered. When the event is triggered, it automatically passes an event object (eventObject) to the listener object method. Each event object has properties that contain information about the event. You can use these properties to write code that handles the event. For more information, see EventDispatcher class (API).

Finally, you call the addEventListener() method on the component instance that broadcasts the event to register the listener with the instance. When the instance dispatches the event, the listener is called.

The second usage example uses an on() handler and must be attached directly to a Gallery instance. The keyword this, used inside an on() handler attached to a component, refers to the component instance.

Example

The following example displays the description of the selected item whenever the front item is clicked. To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

var galleryListener:Object = new Object();
galleryListener.click = function(evt_obj:Object) {
    trace(evt_obj.target.selectedItem.description);
};
My_gallery.addEventListener("click", galleryListener);
```

Gallery.complete

Usage

Usage 1:

```
var listenerObject:Object = new Object();
listenerObject.complete = function(eventObject:Object) {
    // Your code here
};
```

Usage 2:

```
on (complete) {
    // Your code here
}
```

Description

Event; triggered when the XML finished loading.

The first usage example uses a dispatcher/listener event model. A component instance dispatches an event (in this case, complete) and the event is handled by a function, also called a handler, on a listener object (listenerObject) that you create. You define a method with the same name as the event on the listener object; the method is called when the event is triggered. When the event is triggered, it automatically passes an event object (eventObject) to the listener object method. Each event object has properties that contain information about the event. You can use these properties to write code that handles the event. For more information, see EventDispatcher class (API).

Finally, you call the addEventListener() method on the component instance that broadcasts the event to register the listener with the instance. When the instance dispatches the event, the listener is called.

The second usage example uses an on() handler and must be attached directly to a Gallery instance. The keyword this, used inside an on() handler attached to a component, refers to the component instance.

Example

The following example loads an xml and when loading is completed the listener displays a message in the Output panel. To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

var galleryListener:Object = new Object();
galleryListener.complete = function(evt_obj:Object) {
    trace("Loading complete");
};
my_gallery.addEventListener("complete", galleryListener);

my_gallery.load("gallery.xml");
```

Gallery.doubleClick

Usage

Usage 1:

```
var listenerObject:Object = new Object();
listenerObject.doubleClick = function(eventObject:Object) {
    // Your code here
};
```

Usage 2:

```
on (doubleClick) {
    // Your code here
}
```

Description

Event; broadcast when front image is double clicked.

The first usage example uses a dispatcher/listener event model. A component instance dispatches an event (in this case, doubleClick) and the event is handled by a function, also called a handler, on a listener object (listenerObject) that you create. You define a method with the same name as the event on the listener object; the method is called when the event is triggered. When the event is triggered, it automatically passes an event object (eventObject) to the listener object method. Each event object has properties that contain information about the event. You can use these properties to write code that handles the event. For more information, see EventDispatcher class (API).

Finally, you call the addEventListener() method on the component instance that broadcasts the event to register the listener with the instance. When the instance dispatches the event, the listener is called.

The second usage example uses an on() handler and must be attached directly to a Gallery instance. The keyword this, used inside an on() handler attached to a component, refers to the component instance.

Example

The following example displays the description of the selected item whenever the front item is double clicked. To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:”me.jpg”, description:”This is me”});
my_gallery.addItem({url:”colleagues.jpg”, description:”My colleagues”});
my_gallery.addItem({url:”office.jpg”, description:”The office”});

var galleryListener:Object = new Object();
galleryListener.doubleClick = function(evt_obj:Object) {
    trace(evt_obj.target.selectedItem.description);
};
My_gallery.addEventListener(“doubleClick”, galleryListener);
```

Gallery.motionFinished

Usage

Usage 1:

```
var listenerObject:Object = new Object();
listenerObject.motionFinished = function(eventObject:Object) {
    // Your code here
};
```

Usage 2:

```
on (motionFinished) {
    // Your code here
}
```

Description

Event; triggered when the tween animation finished.

The first usage example uses a dispatcher/listener event model. A component instance dispatches an event (in this case, motionFinished) and the event is handled by a function, also called a handler, on a listener object (listenerObject) that you create. You define a method with the same name as the event on the listener object; the method is called when the event is triggered. When the event is triggered, it automatically passes an event object (eventObject) to the listener object method. Each event object has properties that contain information about the event. You can use these properties to write code that handles the event. For more information, see EventDispatcher class (API).

Finally, you call the addEventListener() method on the component instance that broadcasts the event to register the listener with the instance. When the instance dispatches the event, the listener is called.

The second usage example uses an on() handler and must be attached directly to a Gallery instance. The keyword this, used inside an on() handler attached to a component, refers to the component instance

Example

The following example displays a message in the Output panel when a motion is finished. To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

my_gallery.addItem({url:"me.jpg", description:"This is me"});
my_gallery.addItem({url:"colleagues.jpg", description:"My colleagues"});
my_gallery.addItem({url:"office.jpg", description:"The office"});

var galleryListener:Object = new Object();
galleryListener.motionFinished = function(evt_obj:Object) {
    trace("Finished with the motion to index " + evt_obj.target.selectedIndex);
};
my_gallery.addEventListener("motionFinished", galleryListener);
```

Gallery.progress

Usage

Usage 1:

```
var listenerObject:Object = new Object();
listenerObject.progress = function(eventObject:Object) {
    // Your code here
};
```

Usage 2:

```
on (progress) {
    // Your code here
}
```

Description

Event; triggered while the XML is loading.

The first usage example uses a dispatcher/listener event model. A component instance dispatches an event (in this case, progress) and the event is handled by a function, also called a handler, on a listener object (listenerObject) that you create. You define a method with the same name as the event on the listener object; the method is called when the event is triggered. When the event is triggered, it automatically passes an event object (eventObject) to the listener object method. Each event object has properties that contain information about the event. You can use these properties to write code that handles the event. For more information, see EventDispatcher class (API).

Finally, you call the addEventListener() method on the component instance that broadcasts the event to register the listener with the instance. When the instance dispatches the event, the listener is called.

The second usage example uses an on() handler and must be attached directly to a Gallery instance. The keyword this, used inside an on() handler attached to a component, refers to the component instance. For example, the following code, attached to the List instance my_gallery, sends "_level0.my_gallery" to the Output panel:

```
on (progress) {
    trace(this);
}
```

Example

The following example creates a listener object with a progress handler that traces the percent loaded and sends it to the output panel. To try this example, drag a Gallery component to the Stage and give it the instance name **my_gallery**. Add the following code to Frame 1 in the timeline:

```
var my_gallery:com.flzone.imageflow.Gallery;

var loadListener:Object = new Object();
loadListener.progress = function(evt_obj:Object) {
    trace("The xml is " + my_gallery.percentLoaded + "% loaded.");
};
my_gallery.addEventListener("progress", loadListener);
my_gallery.load("gallery.xml");
```