

3D Sphere

User Guide revision 1.2
www.flashloaded.com

Table of Contents

Installation	4
Getting started	5
XML	6
Setting component parameters in the XML file	7
Component Inspector Parameters	8
General settings	8
Sphere settings	8
Thumbnail settings	9
Large image settings	10
Mouse interactivity settings	10
Opening animation settings	11
Camera settings	11
Animation settings	12
Enabling mouse scrolling for Mac browsers	13
How to improve performance	14
Resizing large images	14
Skinning	15
Skinning the preloader animation	15
Skinning the preloader textfield	16
Skinning the tooltip	16
Skinning the backs of thumbnails	16
ActionScript events	17

Adding descriptions to images	18
Opening a URL on click	19
ActionScript properties	20
ActionScript methods	28
Help	29

Installation

You will need Adobe Extension Manager in order to install this component. Extension Manager should have been installed by default when you installed Flash. You may download the latest version of Extension Manager for free from the [Adobe website](#).

1. Ensure that Flash is closed before installing the 3D Sphere component.
2. Unzip/extract the 3DSphere.zip file that you downloaded. You will find a file called 3DSphere.mxp. Double click on this file in order to install the component using Extension Manager.

The 3D Sphere should now be installed in your Flash Components Panel.

Getting started

1. Having installed the 3D Sphere using the Adobe Extension Manager, start a new Flash ActionScript 3 file and save it.
2. Prepare your images:
The 3D Sphere uses different images for the large and thumbnail images. In order to set this up, create two folders called, for example, *thumbs* and *images* in the same location as your Flash file. Place the thumbnail images in the thumbs folder and the large images in the images folder.
3. At this stage, you should create the XML file which contains the image data. Please refer to the [XML](#) section of this userguide for instructions on creating the XML file.
4. Locate the 3D Sphere component in the Components panel and drag it onto the stage.
5. Use the Free Transform tool or the properties panel to resize the component to the desired display area.
6. Click on the component and open the Component Inspector panel (shift +F7).
7. Enter the name of the XML file that you created in step 3 in the XMLSource parameter.
8. At this stage, you can already test the 3D Sphere with the default parameters, to ensure that you have set it up correctly. Press *Ctrl+Enter* (win) or *Cmd+Enter* (mac) to test your movie.
9. You can change the various parameter settings in the Component Inspector to obtain the desired look. Please see the [Component Inspector parameters](#) section for a description of each setting.

Note: In order for the animations to be smooth it's recommended to set your movie speed to 31 fps.

XML

All of the images for the 3D Sphere are specified using an XML file. You can also set all of the parameters in the XML file. By defining the images and parameters in an external XML file, you can publish the SWF file once and change the images or parameters whenever you wish.

Note: The width and height values should be declared for all images, unless the images are exactly the same size, in which case they will all use the defaultImageHeight and defaultImageWidth settings.

1. Open your favorite plain text editor (for example Notepad on Windows or TextEdit on Mac) and start a new file. *Note: If you are using TextEdit on Mac, choose Format > Make Plain Text*
2. Begin your file with the following line:

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

This is the standard xml declaration.

3. Add the following lines to your xml file (the bold lines are the new additions)

```
<?xml version="1.0" encoding="utf-8"?>  
<gallery>  
</gallery>
```

5. Define the images within the *gallery* tags (the bold lines are the new additions)

```
<?xml version="1.0" encoding="utf-8"?>  
<gallery>  
  <image src="images/img1.jpg" thumbnailSrc="thumbs/img1.jpg"  
width="520" height="345" tooltip="Tree" description="Desc 1" />  
  <image src="images/img2.jpg" thumbnailSrc="thumbs/img2.jpg"  
width="520" height="345" tooltip="House" description="Desc 2" />  
</gallery>
```

src defines path and name an large image.

thumbnailSrc defines path and name an thumbnail image.

width (optional) overrides the defaultImageWidth setting for the individual image.

height (optional) overrides the defaultImageHeight setting for the individual image.

tooltip (optional) the tooltip text for the thumbnail.

description (optional) an optional description which can be read and displayed using ActionScript.

6. Save the XML file to the same folder as your Flash file. In this example, we have given the XML file the name: *images.xml*

7. Return to your Flash file. Enter the name and path to the XML file that you just created in the **XMLSource** parameter of the 3D Sphere that's on the stage.

Note: If your .swf file will be in a different folder to the HTML file in which it is embedded, you should enter the path to the XML file, relative to the location of the .html file.

8. Press Ctrl+Enter (Win) or Cmnd+Enter (Mac) to test your movie.

Setting component parameters in the XML file

All of the parameters that appear in the Component Inspector can be set in the XML file. Any parameter that is set in the XML will override the value for that parameter that has been set in the Component Inspector.

In order to set a parameter in the XML file, simply define the parameter and the value in the *gallery* tag like this:

```
<gallery radius="700" mirrorThumb="true" zoomBehaviour="zoomThumb" >
```

Component Inspector Parameters

General settings

Parameter	Description	Example
XMLSource	The path and filename of the XML file containing the image information.	images.xml
preloadText	The text that should appear to indicate the number of thumbnails loading and total thumbnails to load. Use %NUM% to represent the number of thumbnails loaded and %TOTAL% to represent the total loaded.	Loaded %NUM% of %TOTAL% thumbnails
showTooltip	Sets whether to display a tooltip on thumbnail mouse over or not.	true
smoothing	Sets whether the smoothing of images is controlled automatically by the 3D Sphere or if smoothing is always on or off. When set in <i>auto</i> mode, smoothing is automatically turned off while animations are occurring. Setting this parameter to <i>on</i> may have an impact on performance on some spheres. Options available are: <i>on</i> , <i>auto</i> or <i>off</i>	auto

Sphere settings

Parameter	Description	Example
radius	The radius of the 3D Sphere	700
segmentsW	Defines the number of segments to divide the sphere into along the width. Setting this at a higher number will make the sphere fuller.	12
segmentsH	Defines the number of segments to divide the sphere into along the height. Setting this at a higher number will make the sphere fuller.	12
transparentSphere	Sets whether the sphere is transparent if it should be the solid color specified in the <i>sphereColor</i> parameter.	true

Parameter	Description	Example
sphereColor	The color of the sphere is <i>transparentSphere</i> is set to <i>false</i> .	#00FF00
sphereQuality	The smoothness quality of the sphere. A higher number will be smoother but also more processor intensive.	64
sphereGlowColor	The color of the optional glow around the sphere.	#FFFFFF
sphereGlowAlpha	The opacity of the glow (between 0 - 1). A value of 0 would show no glow.	0.5
sphereGlowBlurX	The X extend of the glow blur (in pixels)	32
sphereGlowBlurY	The Y extend of the glow blur (in pixels)	32
sphereGlowStrength	The intensity of the glow	2
sphereGlowQuality	The quality of the glow	1

Thumbnail settings

Parameter	Description	Example
thumbWidth	The width of the thumbnail images	80
thumbHeight	The height of the thumbnail images	60
keepThumbProportion	Sets whether the thumbnails should maintain the proportions of the large image or if they should be resized disproportionately to all be of the same size.	true
repeatThumbs	Sets whether to repeat the thumbnail images in order to create a full sphere or not. If set to <i>false</i> , only the exact number of defined thumbnails will be displayed.	true
mirrorThumb	Sets whether to display a mirrored copy of the thumbnail image as the back of the thumbnail or not. If set to <i>false</i> , the <i>Back of thumb</i> movie clip will be used instead. See the skinning section for instructions to skin the <i>Back of thumb</i> movie clip.	true
backThumbBrightness	The brightness of the thumbnails when viewed from behind.	-120

Parameter	Description	Example
displaySingleImageTop	Sets whether to display a single thumbnail on the top of the sphere or not.	true
displaySingleImageBottom	Sets whether to display a single thumbnail on the bottom of the sphere or not.	true

Large image settings

Parameter	Description	Example
defaultImageHeight	The default height of the large images. This value can be overridden for individual images in the XML.	450
defaultImageWidth	The default width of the large images. This value can be overridden for individual images in the XML.	600
scaleImageDown	Sets whether images that are too large to fit into the viewport of the component are scaled down proportionally to fit.	true

Mouse interactivity settings

Parameter	Description	Example
mouseSpeed	Sets the speed at which the 3D Sphere rotates when moving the mouse.	20
wheelEnabled	Sets whether to enable mouse scroll wheel zooming or not.	true
xAxisEnabled	Sets whether the sphere can rotate around the X-Axis or not.	true
yAxisEnabled	Sets whether the sphere can rotate around the Y-Axis or not.	true
clickToSpin	Sets whether the sphere spins so that the clicked thumbnail rotates to the center of the component.	true

Parameter	Description	Example
mouseDrag	Sets whether the sphere rotates by clicking and dragging the mouse or by free mouse movement.	true
deceleration	The speed at which the sphere slows down when clicking and dragging if <i>mouseDrag</i> is set to <i>true</i> .	0.3

Opening animation settings

Parameter	Description	Example
openingAnimation	Select whether the thumbnails should animate in from the <i>side</i> or the <i>center</i> .	side
sideAnimationInitialRadius	The radius from which the thumbnails should start appearing when setting <i>openingAnimation</i> to <i>side</i> .	900
openingAnimationFunc	The easing effect to use for the opening animation.	easeInSine
openingAnimationTime	The speed (in seconds) at which the opening animation should occur.	2

Camera settings

Parameter	Description	Example
cameraDistance	The distance of 3D Sphere from the camera.	2000
initialVerticalCameraAngle	The opening angle of the camera, when the sphere is loaded.	0
keepLightSourceInCenter	Sets whether to keep the light source on the colored sphere in the center or to rotate the light source with the sphere, causing the other side to be in shadow.	true

Animation settings

Parameter	Description	Example
clickFollowedByZoom	If set to <i>true</i> , the sphere will rotate the clicked thumbnail to the center and then it will zoom in to display large image.	false
displacement	The number of pixels that the thumbnail should move forward on mouse over.	50
displacementAnimFunc	The easing effect to use for the displacement animation.	easeInSine
displacementAnimTime	The speed (in seconds) at which the displacement animation should occur.	0.4
zoomBehaviour	Sets whether to slide the camera closer in to view the large image (<i>slideCamera</i>) or whether to zoom the thumbnail out of the sphere (<i>zoomThumb</i>).	slideCamera
zoomAnimFunc	The easing effect to use for the opening and closing the large images.	easeInSine
zoomAnimTime	The speed (in seconds) at which the large image animation should occur.	1
showHideAnimFunc		
showHideAnimTime		

* The following easing styles are available:

linear, easeInQuad, easeOutQuad, easeInOutQuad, easeInExpo, easeOutExpo, easeInOutExpo, easeOutInExpo, easeInElastic, easeOutElastic, easeInOutElastic, easeOutInElastic, easeInBack, easeOutBack, easeInOutBack, easeOutInBack, easeOutBounce, easeInBounce, easeInOutBounce, easeOutInBounce, easeInCubic, easeOutCubic, easeInOutCubic, easeOutInCubic, easeInQuart, easeOutQuart, easeInOutQuart, easeOutInQuart, easeInQuint, easeOutQuint, easeInOutQuint, easeOutInQuint, easeInSine, easeOutSine, easeInOutSine, easeOutInSine, easeInCirc, easeOutCirc, easeInOutCirc, easeOutInCirc

Enabling mouse scrolling for Mac browsers

The current version of the Flash Player does not natively support mouse wheel scrolling in Mac browsers. We have built a solution for this into the 3D Sphere. In order to use this solution, you must construct your HTML file like this:

1. Copy the **js** folder, that was included with your download, to the same folder in which your HTML file will reside.
2. Write the following code in the <head></head> section of your HTML file:

```
<script type="text/javascript" src="js/swfobject.js"></script>
<script type="text/javascript" src="js/swfmacmousewheel2.js"></script>
<script type="text/javascript">
  var vars = {};
  var params = { scale:'noScale', salign:'lt', menu:'true' };
    var attributes = { id:'sphereObject', name:'sphereObject' };
    swfmacmousewheel.registerObject(attributes.id);
</script>
```

3. Write the following code in the body of your HTML file, where you would like the Flash SWF to be located:

```
<script>swfobject.embedSWF("sphereexample.swf", "flashContent",
"1000", "600", "9.0.0", "js/expressInstall.swf", vars, params,
attributes );</script>
```

Note: Change the items marked in bold to match your SWF filename, height and width.

This will work when testing online only. You must ensure that you upload the **js** folder with your HTML file.

How to improve performance

There are several factors which can affect the performance of the 3D Sphere, especially when viewed on older computers. The following settings will help improve the performance:

1. Set ***smoothing*** to *automatic* or *false*
2. If ***sphereQuality*** to a lower number (try set it to 32)
3. Large component sizes can impact performance, especially if there are a large number of images. Try make the component size smaller.

Note: You may need to tweak the above settings based on the number of images displayed.

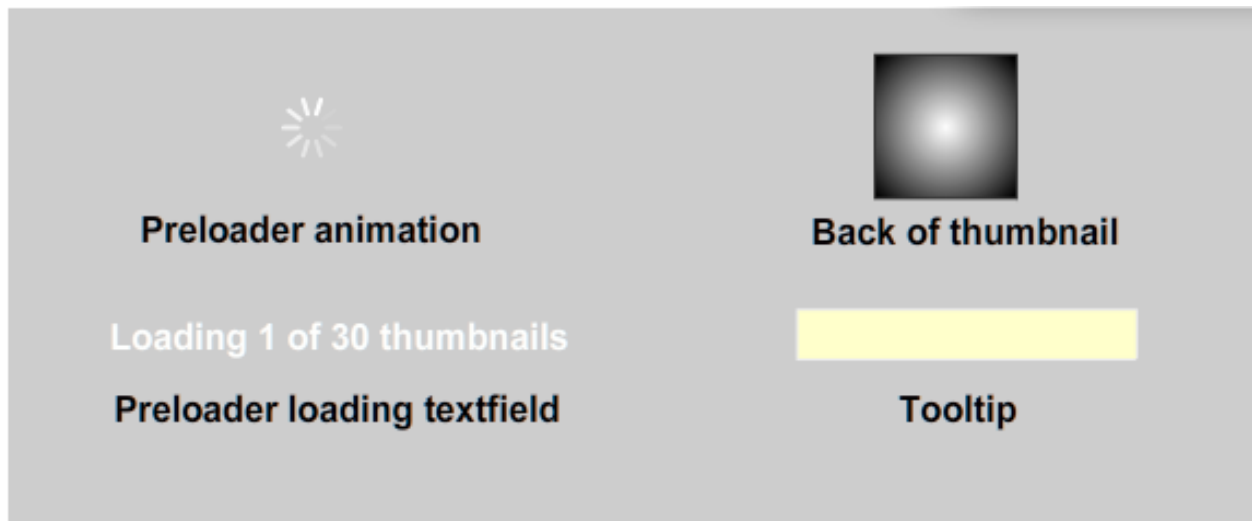
Resizing large images

The 3D Sphere does not resize the large images by specifying a smaller image size, although the size of the large images must be specified in the component or XML file. This size specification is necessary in order for the 3D Sphere to zoom to the correct size and to resize the thumbnails proportionally. The large images can be automatically scaled down to fit the maximum size of the component by setting the *scaleImageDown* parameter to *true*.

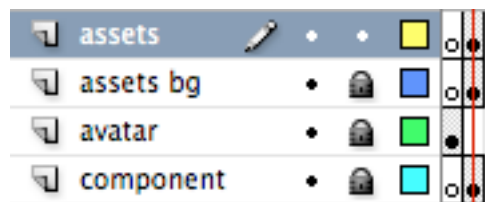
Skinning

The preloader animation, preloader text style, tooltip and back of thumbnails can be skinned to match your desired look and feel. Double click anywhere on the 3D Sphere component that's on the stage in order to skin the elements.

You should now see the skinnable movie clips:



Note: In order to skin any of these elements, you must unlock the assets layer:



Skinning the preloader animation

You can change the color of the built-in preloader animation by changing the tint of the preloader movie clip or you can double click on this movie clip in order to change the symbols and animation entirely.

Note: You can replace this movie clip with another animation if you wish. The preloader animation movie clip must have a center registration point.

Skinning the preloader textfield

You can change the font style, size and color of the preloading text by editing the textfield inside the **thumbsProgress** movie clip.

Skinning the tooltip

Double click on the **Gallery_tooltip** movie clip in order to edit the look of the tooltip. You can change the font and style of the text as well as the background.

Note: The tooltip uses 9-slice scaling in order to resize dynamically correctly. Move the grid to ensure that the parts that are in the 4 corners are the parts that will not resize.

Skinning the backs of thumbnails

Double click on the **Gallery_BackThumbSkin** movie clip in order to edit the appearance of the backs of the thumbnails in the event that you have chosen not to display mirrored thumbnails.

ActionScript events

Events are called whenever the 3D Sphere performs the specified action. The component includes an event class called **Sphere3DEvent** in the com.flashloaded.sphere3d package.

The event holds the following image properties:

type: the type of event that was initiated (e.g: *imageSelected*, *imageShown*, *ballShown*)

imageHolder: contains all of the other parameters specified for the image

The following events are included:

Sphere3DEvent.IMAGE_SELECTED

Broadcasted when a thumbnail is clicked

Sphere3DEvent.IMAGE_SHOWN

Broadcasted when a large image has finished loading

Sphere3DEvent.BALL_SHOWN

Broadcasted the large image has been clicked on to return to the sphere mode

Example 1- This outputs a trace for the IMAGE_SELECTED, IMAGE_SHOWN and BALL_SHOWN events:

```
import com.flashloaded.sphere3d.Sphere3DEvent;

3DSphereInstance.addEventListener(Sphere3DEvent.IMAGE_SELECTED,
traceMe);
3DSphereInstance.addEventListener(Sphere3DEvent.IMAGE_SHOWN, traceMe);
3DSphereInstance.addEventListener(Sphere3DEvent.BALL_SHOWN, traceMe);

function traceMe(e:Sphere3DEvent):void {
    trace("action: " + e.type + ", image description: " +
        e.imageHolder.description + ",
        tooltip: " + e.imageHolder.tooltip);
}
```

Example 2- This is how you could use the IMAGE_SELECTED event:

```
import com.flashloaded.sphere3d.Sphere3DEvent;
3DSphereInstance.addEventListener(Sphere3DEvent.IMAGE_SELECTED,clickHa
ndler);

function clickHandler(evt:Sphere3DEvent):void{
    gotoAndStop(evt.imageHolder.param);
}
```

Adding descriptions to images

You can add a description for each image which appears in a textfield. This is done by specifying the title in the ***description*** element of the XML file and by reading the description in the ***IMAGE_SELECTED*** event. This is how you would do this:

1. Add a description value to image in the XML containing the description. For example:

```

```

2. Place a textfield on your stage where you would like the descriptions to appear. Set the textfield to be dynamic and give it an instance name, e.g: *desc_txt*

3. Give the 3D Sphere that's on the stage an instance name, e.g: *sphere*

4. Type the following ActionScript code on the timeline, in the first frame in which the 3D Sphere appears:

```
import com.flashloaded.sphere3d.Sphere3DEvent;
3DSphereInstance.addEventListener(Sphere3DEvent.IMAGE_SHOWN,
selectedHandler);
3DSphereInstance.addEventListener(Sphere3DEvent.BALL_SHOWN,
deselectedHandler);

function selectedHandler(evt:Sphere3DEvent):void{
    desc_txt.text = evt.imageHolder.description;
}

function deselectedHandler(evt:Sphere3DEvent):void{
    desc_txt.text = "";
}
```

*Note: In this example, you would replace **sphere** with the instance name of your 3D Sphere and replace **desc_txt** with the instance name of your description textfield.*

You should now see the description appear when viewing the large images.

Opening a URL on click

You can have a URL open when clicking on an image. This is done by adding a parameter to the *img* tag for each image in the XML file and by reading the URL parameter in the **IMAGE_SELECTED** event. This is how you would do this:

1. Add a value call *url* to image in the XML. The url param will contain the URL. For example:

```

```

2. Give the 3D Sphere that's on the stage an instance name, e.g: *sphere*

4. Type the following ActionScript code on the timeline, in the first frame in which the 3D Sphere appears:

```
import com.flashloaded.sphere3d.Sphere3DEvent;
import flash.net.navigateToURL;
import flash.net.URLRequest;

3DSphereInstance.addEventListener(Sphere3DEvent.IMAGE_SELECTED,
selectedHandler);

function selectedHandler(evt:Sphere3DEvent):void{
    var url:String = evt.imageHolder.url;
    var request:URLRequest = new URLRequest(url);
    navigateToURL(request, "_blank");
}
```

*Note: In this example, you would replace **sphere** with the instance name of your 3D Sphere.*

You should now see that the URL opens when clicking on a thumbnail that has a URL assigned to it.

ActionScript properties

backThumbBrightness:int

Availability

Flash Player 9

Description

Property; brightness of the thumbnails when viewed from behind.

Example

```
3DSphereInstance.backThumbBrightness = -120;
```

cameraZoom:Number

Availability

Flash Player 9

Description

Property; default height of the large images.

Example

```
3DSphereInstance.cameraZoom = 1000;
```

displacementAnimFunc:String

Availability

Flash Player 9

Description

Property; easing effect to use for the displacement animation.

Example

```
3DSphereInstance.displacementAnimFunc = "easeInSine";
```

displacementAnimTime:Number

Availability

Flash Player 9

Description

Property; speed (in seconds) at which the displacement animation should occur.

Example

```
3DSphereInstance.displacementAnimTime = 1;
```

keepThumbProportion:Boolean

Availability

Flash Player 9

Description

Property; sets whether the thumbnails should maintain the proportions of the large image or if they should be resized disproportionately to all be of the same size.

Example

```
3DSphereInstance.keepThumbProportion = true;
```

mirrorThumb:Boolean

Availability

Flash Player 9

Description

Property; sets whether to display a mirrored copy of the thumbnail image as the back of the thumbnail or not. If set to false, the Back of thumb movie clip will be used instead. See the skinning section for instructions to skin the *Back of thumb* movie clip.

Example

```
3DSphereInstance.mirrorThumb = true;
```

mouseSpeed:int

Availability

Flash Player 9

Description

Property; speed at which the 3D Sphere rotates when moving the mouse.

Example

```
3DSphereInstance.mouseSpeed = 20;
```

openingAnimation:String

Availability

Flash Player 9

Description

Property; sets whether the thumbnails should animation in from the the *side* or the *center*.

Example

```
3DSphereInstance.openingAnimation = "side";
```

openingAnimationFunc:String

Availability

Flash Player 9

Description

Property; easing effect to use for the opening animation.

Example

```
3DSphereInstance.openingAnimationFunc = "easeInSine";
```

openingAnimationTime:int

Availability

Flash Player 9

Description

Property; speed (in seconds) at which the opening animation should occur.

Example

```
3DSphereInstance.openingAnimationTime = 1;
```

radius:int

Availability

Flash Player 9

Description

Property; radius of the sphere

Example

```
3DSphereInstance.radius = 700;
```

repeatThumbs:Boolean

Availability

Flash Player 9

Description

Property; sets whether to repeat the thumbnail images in order to create a full sphere or not. If set to *false*, only the exact number of defined thumbnails will be displayed.

Example

```
3DSphereInstance.repeatThumbs = true;
```

segmentsH:int

Availability

Flash Player 9

Description

Property; the number of segments to divide the sphere into along the height. Setting this at a higher number will make the sphere fuller.

Example

```
3DSphereInstance.segmentsH = 12;
```

segmentsW:int

Availability

Flash Player 9

Description

Property; the number of segments to divide the sphere into along the width. Setting this at a higher number will make the sphere fuller.

Example

```
3DSphereInstance.segmentsW = 12;
```

selectedIndex:int

Availability

Flash Player 9

Description

Property; the index of the selected image

Example

```
3DSphereInstance.selectedIndex = 5;
```

showHideAnimFunc:String

Availability

Flash Player 9

Description

Property; easing function for show/hide animation Possible values: linear,easeInSine, easeOutSine, easeInQuad, easeOutQuad, easeInCubic, easeOutCubic, easeInElastic, easeOutElastic, easeInBack, easeOutBack, easeInBounce, easeOutBounce

Example

```
3DSphereInstance.showHideAnimFunc = "easeInSine";
```

showHideAnimTime:Number

Availability

Flash Player 9

Description

Property;

Example

```
3DSphereInstance.showHideAnimTime = 1;
```

showTooltip:Boolean

Availability

Flash Player 9

Description

Property; sets whether to display a tooltip on thumbnail mouse over or not.

Example

```
3DSphereInstance.showTooltip = true;
```

thumbHeight:int

Availability

Flash Player 9

Description

Property; height of the thumbnail images

Example

```
3DSphereInstance.thumbHeight = 20;
```

thumbWidth:int

Availability

Flash Player 9

Description

Property; width of the thumbnail images

Example

```
3DSphereInstance.thumbWidth = 15;
```

thumbLoaderTextPattern:String

Availability

Flash Player 9

Description

Property; text that should appear to indicate the number of thumbnails loading and total thumbnails to load. Use %NUM% to represent the number of thumbnails loaded and %TOTAL% to represent the total loaded.

Example

```
3DSphereInstance.totalColumns = "Loading %NUM% of %TOTAL% thumbs";
```

tooltipShift:int

Availability

Flash Player 9

Description

Property; the number of pixels to offset the tooltip from the mouse pointer

Example

```
3DSphereInstance.totalColumns = 5;
```

wheelEnabled:Boolean

Availability

Flash Player 9

Description

Property; sets whether to enable mouse scroll wheel zooming or not.

Example

```
3DSphereInstance.zoom = true;
```

xAxisEnabled:Boolean

Availability

Flash Player 9

Description

Property; sets whether the sphere can rotate around the X-Axis or not.

Example

```
3DSphereInstance.totalColumns = true;
```

yAxisEnabled:Boolean

Availability

Flash Player 9

Description

Property; sets whether the sphere can rotate around the Y-Axis or not.

Example

```
3DSphereInstance.zoom = true;
```

zoomAnimFunc:String

Availability

Flash Player 9

Description

Property; easing effect to use for the opening and closing the large images. Possible values: linear, easeInSine, easeOutSine, easeInQuad, easeOutQuad, easeInCubic, easeOutCubic, easeInElastic, easeOutElastic, easeInBack, easeOutBack, easeInBounce, easeOutBounce

Example

```
3DSphereInstance.totalColumns = "easeInSine";
```

zoomAnimTime:Number

Availability

Flash Player 9

Description

Property; speed (in seconds) at which the large image animation should occur.

Example

```
3DSphereInstance.zoom = 1;
```

zoomBehaviour:String

Availability

Flash Player 9

Description

Property; set to *slideCamera* or *zoomThumb*.

Example

```
3DSphereInstance.zoom = "slideCamera";
```

ActionScript methods

getImageHolder

Availability

Flash Player 9

Description

Method; returns the ImageHolder object for the plane

Example

```
3DSphereInstance.getImageHolder(plane:DisplayObject3D);
```

openImage

Availability

Flash Player 9

Description

Method; opens the selected image

Example

```
3DSphereInstance.openImage(plane:DisplayObject3D);
```

Help

This component is fully supported by the Flashloaded support team through our support forum. You will also find tips and additional information in the forum as well as announcements for version updates: [3D Sphere Support Forum](#)

Note: In order to post a question in the forum, you will need to [register](#) by creating a username and password. This registration differs from your account login.