

3D Tunnel

User Guide revision 1.0
www.flashloaded.com

Table of Contents

Installation	3
Getting started	4
XML	5
Setting component parameters in the XML file	7
Component Inspector Parameters	8
General settings	8
Camera & motion settings	8
Thumbnail settings	9
Preloader settings	9
Opening animation	10
Enabling mouse scrolling for Mac browsers	11
Resizing large images	12
Skinning	13
Skinning the main preloader	14
Skinning the large image preloader	14
ActionScript events	15
Adding titles to images	17
Opening a URL on click	18
ActionScript methods	19
Help	21

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 Tunnel component.
2. Unzip/extract the 3DTunnel.zip file that you downloaded. You will find a file called 3DTunnel.mxp. Double click on this file in order to install the component using Extension Manager.

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

Getting started

1. Having installed the 3D Tunnel using the Adobe Extension Manager, start a new Flash ActionScript 3 file and save it.
2. Prepare your images:
Place the thumbnail and large images in the same or different folders, in the same location as your Flash file.
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 Tunnel 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 Tunnel with the default parameters, to ensure that you have set it up correctly. Press Ctrl+Enter (win) or Cmnd+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 on 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 Tunnel 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.

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 xmlns:media="http://search.yahoo.com/mrss/">  
</gallery>
```

4. Add the item and media tags to your XML file (the bold lines are the new additions)

```
<?xml version="1.0" encoding="utf-8"?>  
<gallery xmlns:media="http://search.yahoo.com/mrss/">  
  <item>  
    <media:content url="images/1.jpg" type="image/jpeg" width="540"  
height="360">  
      <media:thumbnail url = "thumbs/1.jpg" />  
    </media:content>  
  </item>  
</gallery>
```

Explanation of the item tag:

Each image is defined within its own *item* tags. The *item* tags contain *media* tags which define each aspect of the image:

media:content - defines the filename (url) of the large image, the image type and the width and height of the large image. The content tag also contains the *text* and *thumbnail* definition tags.

media:thumbnail - defines the filename (url) of the thumbnail image.

Example of XML file with 3 images defined:

```
<?xml version="1.0" encoding="utf-8"?>
<gallery xmlns:media="http://search.yahoo.com/mrss/">
  <item>
    <media:content url="images/1.jpg" type="image/jpeg" width="540"
height="360">
      <media:thumbnail url = "thumbs/1.jpg"/>
    </media:content>
  </item>

  <item>
    <media:content url="images/2.jpg" type="image/jpeg" width="540"
height="360">
      <media:thumbnail url = "thumbs/2.jpg"/>
    </media:content>
  </item>

  <item>
    <media:content url="images/3.jpg" type="image/jpeg" width="540"
height="360">
      <media:thumbnail url = "thumbs/3.jpg"/>
    </media:content>
  </item>
</gallery>
```

5. Each image can also contain 3 optional additional parameters for title, description and any other value (e.g: URL, frame number). These parameters can be read through ActionScript events. These parameters are added to the images like this:

```
<item>
  <media:content url="images/3.jpg" type="image/jpeg" width="540"
height="360">
    <media:thumbnail url = "thumbs/3.jpg"/>
    <media:title>My title</media:title>
    <media:desc>My Description</media:desc>
    <media:param>http://www.flashloaded.com</media:param>
  </media:content>
</item>
```

5. 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*

6. 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 Tunnel 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.

7. 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 xmlns:media="http://search.yahoo.com/mrss/" totalImages="40">
```

Component Inspector Parameters

General settings

Parameter	Description	Example
XMLSource	The path and filename of the XML file containing the image information.	images.xml
totalImages	The total number of thumbnails to display. If a larger number is specified than the amount of thumbnails defined in the XML file, the images are repeated.	35
numSides	Defines the number of sides of the tunnel. This should never be set below 2.	5
spaceBetweenSides	Defines the number of pixels between the sides. Note: Setting this to 0 will still show a small gap.	
spaceBetweenThumbs	Defines the number of pixels between the thumbnails on each side.	
smoothImages	Sets whether to switch image smoothing on (true) or off (false).	true
useKeyboardArrows	Sets whether to enable keyboard arrow keys to zoom in and out of the tunnel.	
useMouseWheel	Sets whether to enable mousewheel scrolling to zoom in and out of the tunnel.	true

Camera & motion settings

Parameter	Description	Example
cameraMotionStep	The number of pixels to move the camera at a time in and out of the tunnel with each up/down arrow key press.	50
cameraStartRotation	Defines the angle at which to rotate the tunnel initially.	30
cameraStartZ	Defines the Z depth at which to display the tunnel.	

Parameter	Description	Example
motionEasing	Defines the easing effect to apply to the motion of zooming in and out of the large image.	easeOutElastic
motionTime	Defines the time taken for the large image zoom in/out motion to complete.	2

Thumbnail settings

Parameter	Description	Example
thumbHeight	Defines the maximum height for the thumbnails.	120
thumbWidth	Defines the maximum width for the thumbnails.	180
closeAndDistantThumbsRatio	Defines the ratio of the height of close to distant thumbnail sizes to create more or less perspective. A higher number = more perspective.	3

Preloader settings

Parameter	Description	Example
loadAll	Sets whether to preload all of the large images before displaying any images or to load large images on demand.	false
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
preloaderAlpha	The percentage of opacity of the preloader.	75

Opening animation

Parameter	Description	Example
startAnimationDelayBetweenRows	Defines the time to wait before displaying the next row of images during the opening animation.	0.5
startAnimationEasing	Defines the easing effect to use for the opening animation.	easeOutBounce
startAnimationOutside	Sets whether the animation starts from outside of the stage area or inside.	true
startAnimationSpinning	Sets whether the opening animation spins or not.	true
startAnimationTime	Defines the time for the opening animation to complete.	3

* 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 Tunnel. 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:'tunnelObject', name:'wallObject' };
    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("tunnelexample.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.

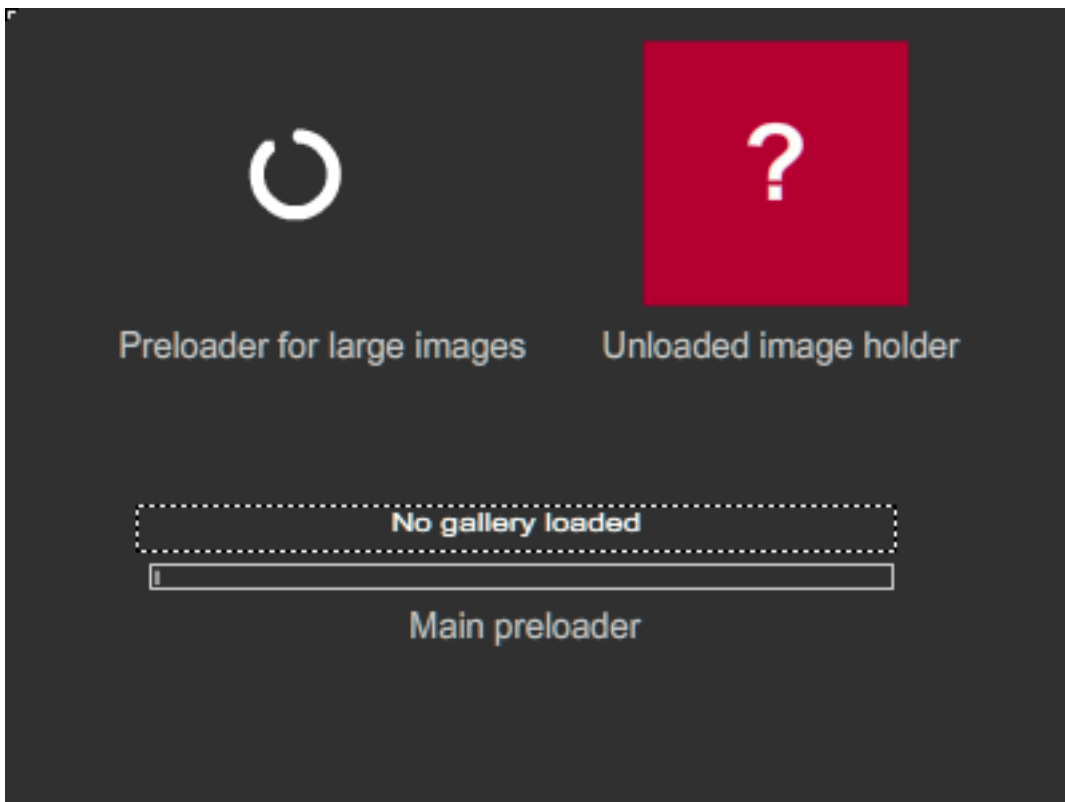
Resizing large images

The 3D Tunnel 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 Tunnel to zoom to the correct size and to resize the thumbnails proportionally. Images that are too large for the component area are automatically scaled down to fit the maximum size of the component.

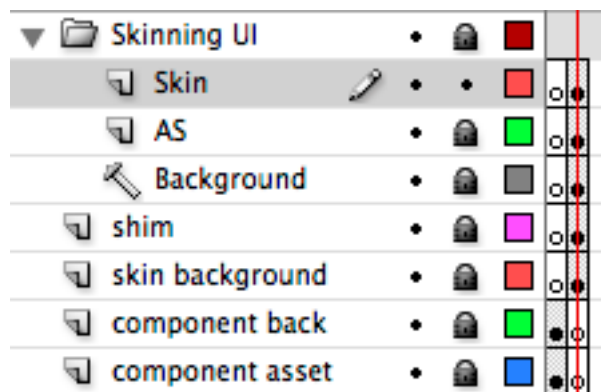
Skinning

The preloader animation, preloader text and missing image symbol can be skinned to match your desired look and feel. Double click anywhere on the 3D Tunnel 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 Skin layer:



Skinning the main preloader

Double click on the **Main preloader** movie clip in order to skin the main opening preloader bar and the textfield for the preloader text. You can also change the length of the preloader progress bar.

Skinning the large image preloader

Double click on the **Preloader for large images (preloader_for_content)** movie clip in order to edit the large image preloader.

ActionScript events

Events are called whenever the 3D Tunnel performs the specified action. The component includes an event class called *GalleryEvent* in the *com.flashloaded.tunnel3d.GalleryEvent* package.

The event has an item property which holds the following image properties:

targetNumber: the number/ID of the selected item.

mediaUrl: the URL of the large image.

mediaType: the type of large image.

mediaWidth: the large image width.

mediaHeight: the large image height.

mediaTitle: the value of the *title* tag for the item.

mediaDesc: the value of the *desc* tag for the item.

mediaParam: the value of the *param* tag for the item.

thumbnailUrl: the URL of the thumbnail image.

The following events are included:

GalleryEvent.ITEM_CLICKED

Broadcasted when the thumbnail is clicked.

GalleryEvent.ITEM_SELECTED

Broadcasted when the large image is shown.

GalleryEvent.ITEM_DESELECTED

Broadcasted when the current image is deselected.

GalleryEvent.ITEM_MOUSE_OVER

Broadcasted when the mouse moves over a thumbnail.

GalleryEvent.ITEM_MOUSE_OUT

Broadcasted when the mouse moves off a thumbnail.

GalleryEvent.LOADED

Broadcasted when all of the thumbnails have finished loading.

GalleryEvent.RENDERING_COMPLETE

Broadcasted when the thumbnails have finished the opening animation.

Example 1- This outputs a trace for the ***ITEM_CLICKED***, ***ITEM_MOUSE_OVER*** and ***ITEM_MOUSE_OUT*** events:

```
import com.flashloaded.tunnel3d.GalleryEvent;

3DTunnelInstance.addEventListener(GalleryEvent.ITEM_CLICKED, traceMe);
3DTunnelInstance.addEventListener(GalleryEvent.ITEM_MOUSE_OVER,
traceMe);
3DTunnelInstance.addEventListener(GalleryEvent.ITEM_MOUSE_OUT, traceMe);

function traceMe(e:GalleryEvent):void {
    trace("title: " + e.mediaTitle + ", description: " + e.mediaDesc +
", param: " + e.mediaParam);
}
```

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

```
import com.flashloaded.tunnel3d.GalleryEvent;
3DTunnelInstance.addEventListener(GalleryEvent.ITEM_CLICKED,clickHandler
);

function clickHandler(e:GalleryEvent):void{
    gotoAndStop(e.mediaParam);
}
```

Adding titles to images

You can add a title or description for each image which appears in a textfield. This is done by adding a parameter containing the title in the XML file and by reading the parameter in the **ITEM_SELECTED** event. This is how you would do this:

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

```
<item>
  <media:content url="images/1.jpg" type="image/jpeg" width="540"
height="360">
    <media:title>Image 1 title</media:title>
    <media:thumbnail url = "thumbs/1.jpg"/>
  </media:content>
</item>
```

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

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

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

```
import com.flashloaded.tunnel3d.GalleryEvent;
tunnel.addEventListener(GalleryEvent.ITEM_SELECTED,selectedHandler);
tunnel.addEventListener(GalleryEvent.ITEM_DESELECTED,deselectedHandler);

function selectedHandler(evt:GalleryEvent):void{
    title_txt.text = evt.mediaTitle;
}

function deselectedHandler(evt:GalleryEvent):void{
    title_txt.text = "";
}
```

*Note: In this example, you would replace **tunnel** with the instance name of your 3D Tunnel and replace **title_txt** with the instance name of your title textfield.*

You should now see that the title appears when viewing and moving between the large images.

Opening a URL on click

You can have a URL open when clicking on an image. This is done by specifying the URL in the **media:param** element of the XML file and by reading the URL in the **ITEM_CLICKED** event. This is how you would do this:

1. Add a parameter called param to each image in the XML. The parameter will contain the URL. For example:

```
<item>
  <media:content url="images/1.jpg" type="image/jpeg" width="540"
height="360">
    <media:param>http://www.flashloaded.com</media:param>
    <media:thumbnail url = "thumbs/1.jpg"/>
  </media:content>
</item>
```

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

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

```
import com.flashloaded.tunnel3d.GalleryEvent;
import flash.net.navigateToURL;
import flash.net.URLRequest;

tunnel.addEventListener(GalleryEvent.ITEM_CLICKED,clickHandler);

function clickHandler(evt:GalleryEvent):void{
    var url:String = evt.mediaParam;
    var request:URLRequest = new URLRequest(url);
    navigateToURL(request, "_blank");
}
```

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

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

ActionScript methods

getClipsNumber

Availability

Flash Player 9

Description

Method; returns the total number of clips used in the 3D Tunnel.

Example

```
3DTunnelInstance.getClipsNumber();
```

goLeft

Availability

Flash Player 9

Description

Method; moves the camera one image to the left of the current image.

Example

```
3DTunnelInstance.goLeft();
```

goRight

Availability

Flash Player 9

Description

Method; moves the camera one image to the right of the current image.

Example

```
3DTunnelInstance.goRight();
```

goTop

Availability

Flash Player 9

Description

Method; moves the camera one image above the current image.

Example

```
3DTunnelInstance.goTop();
```

goBottom

Availability

Flash Player 9

Description

Method; moves the camera one image below the current image.

Example

```
3DTunnelInstance.goBottom();
```

gotoImage

Availability

Flash Player 9

Description

Method; moves the camera to the image number specified.

Example

```
3DTunnelInstance.gotoImage(3);
```

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 Tunnel 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.