360PanViewerPRO AS3

User Guide revision 1.0

www.flashloaded.com

Table of Contents

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 360PanViewerPRO-AS3 component.

2. Unzip/extract the 360PanViewerPRO-AS3.zip file that you downloaded. You will find a file called 360PanViewerPRO-AS3.mxp. Double click on this file in order to install the component using Extension Manager.

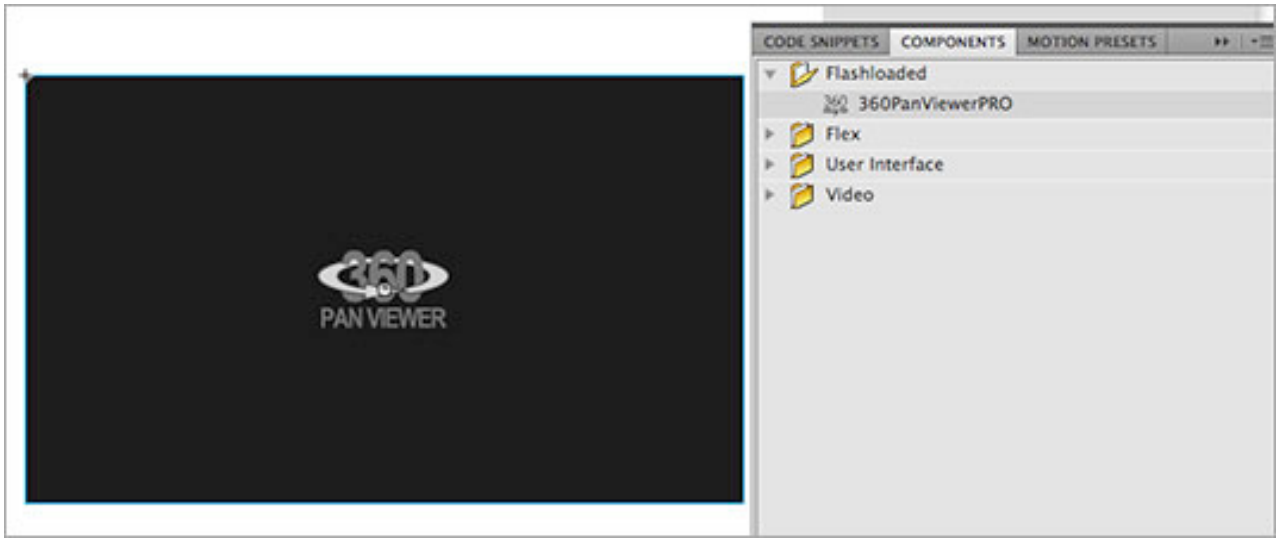
360PanViewerPRO-AS3 should now be installed in your Flash Components Panel.

Getting started

1. Prepare the panoramic image or Flash SWF file that you wish to pan.

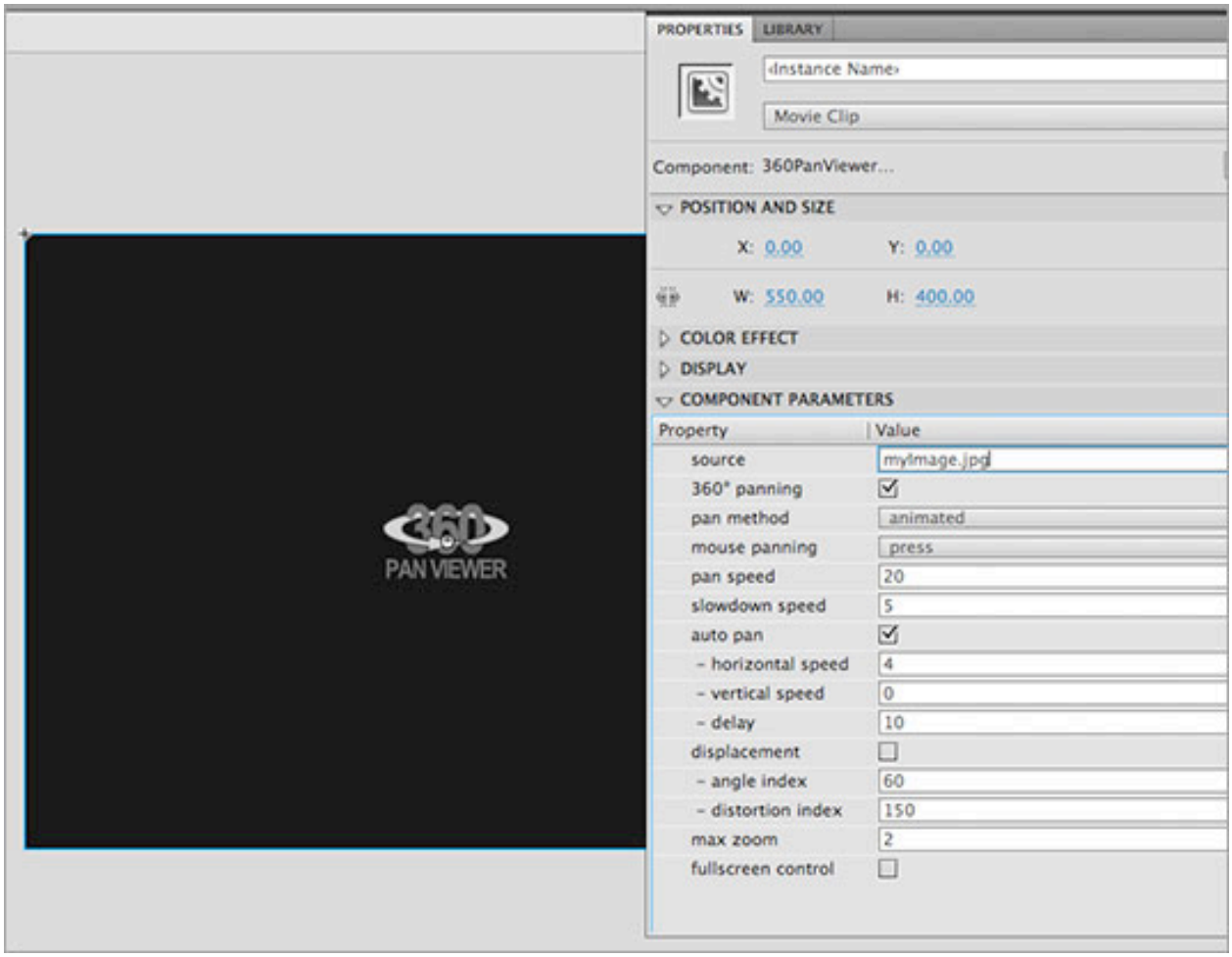
2. Having installed 360PanViewerPRO-AS3 using the Adobe Extension Manager, start a new Flash ActionScript 3.0 file and save it.

3. Locate the ***Flashloaded*** folder in the Components panel and double click on it to expand it. You will find the ***360PanViewerPRO*** component inside this folder.



4. Drag and drop the ***360PanViewerPRO*** component onto the stage and resize it to the desired size using the *Transform* tool. If you make the component larger than the image width, the component will automatically resize, using the image width as the display width.

5. Click on the 360PanViewerPRO component that's on the stage and open the Component Inspector panel (shift +F7). (If you are using Flash CS5 you will find the Component Inspector parameters have been moved to the Properties panel.)



6. Enter the name and path to the external panoramic image or SWF to display in the ***source*** parameter. If the image resides in the same Flash library, enter the Class name of the image/movie clip instead. If you prefer to specify the image details in an XML file, enter the xml filename and path in the ***source*** parameter.

7. At this stage, you can already test 360PanViewerPRO with the default parameters, to ensure that you have set it up correctly. Press Ctrl+Enter (win) or Cmnd+Enter (mac) to test your project.

8. You can change the various parameter settings in the Component Inspector to obtain the desired look and feel. Please see the Component Inspector parameters section for a description on each setting. See the section called Adding hotspots for instructions on adding hotspots to your images.

Component Inspector parameters

Parameter	Description	Example
source	The pan image to use. This can be either the path and filename to an external image or SWF file, the Class name of an image of movie clip that's in the same library or the path and filename of an external XML file which contains the image data.	pano.jpg
360° panning	Set to *true* if the panned object is a full 360° image. Set to *false* for 180° panning.	true
pan method	The panning method to use: *bitmap*: Allows for panning with distortion and allows for images that are wider than 2880px. Animated movie clip hotspots cannot be used. *animated*: Animated hotspots can be used.	animated
mouse panning	Indicates if panning will occur when the mouse is: over the panning area ("over"), pressed over the panning area ("press") or if the mouse does not control the panning ("none").	press
pan speed	The speed of the pan movement.	23
slowdown speed	The slowdown speed, when the mouse moves out of the panning area.	10
auto pan	Set to *true* to enable auto panning.	true
- horizontal speed	Horizontal speed for auto panning.	4
- vertical speed	Vertical speed for auto panning.	8
- delay	Time delay, in seconds, to wait for auto panning to start.	1
displacement	Set to true to enable image displacement. Note: Displacement will only work when *pan method* is set to *bitmap*.	true
- angle index	Size of the "flat" area of the displacement map.	60

Parameter	Description	Example
- distortion index	Amount of edge distortion for image displacement.	120
max zoom	The maximum zoom magnification permitted.	2
fullscreen control	Boolean value indicating if full screen control is visible (true) or not (false).	false

Defining hotspots

Hotspots are a way to define areas of the image as buttons that can have mouse over or click events. This can be used, for example, to provide more information to users when mousing over various elements in your image, or to link elements in the image to other URL's or frames.

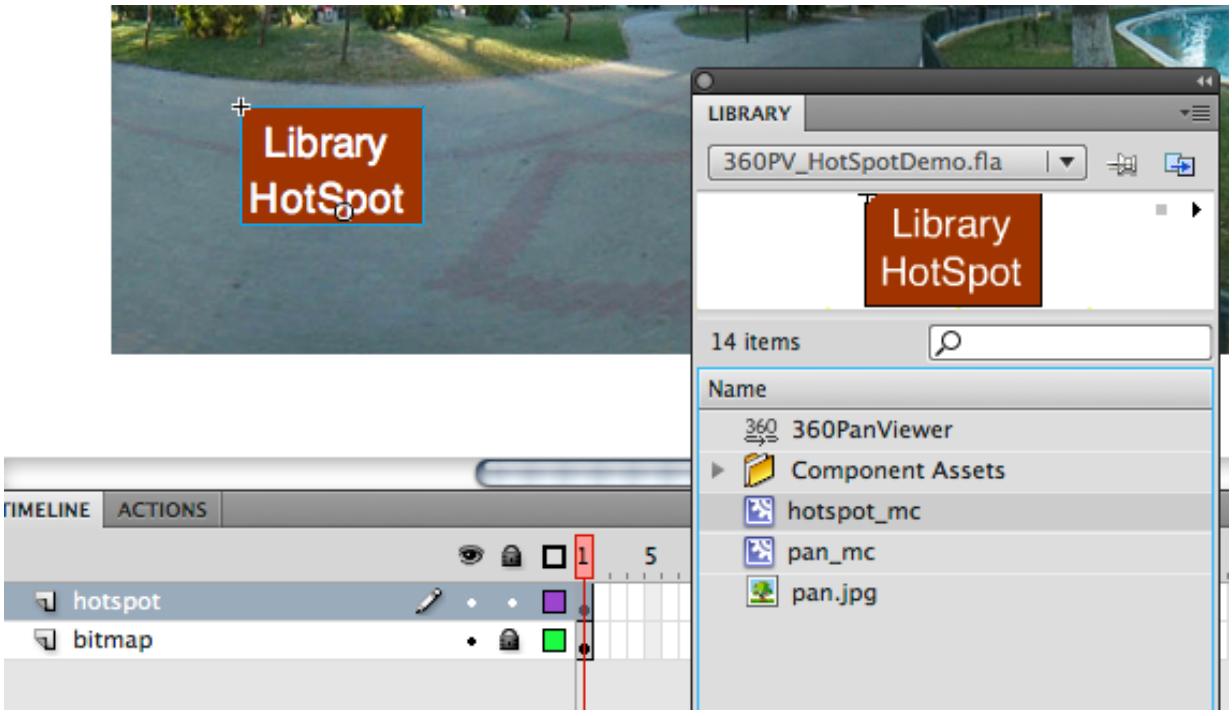
The simplest way to define hotspots is to place them directly over the image in the Flash authoring environment. This method will only work if the image resides in the Flash library (this is not for external images).

Hotspots can also be defined by specifying the coordinates of the hotspots in the XML file (when loading the image through XML), or through ActionScript. Please see the Defining an image through XML section for instructions on adding hotspots through XML.

Placing hotspots directly on the image

1. Create a movie clip containing the image (or vector graphics) that you wish to pan.

2. Add a new layer above the layer containing the image and add movieclips to this layer, over the areas of the image that you want to make hotspots. These movie clips can also be animated (for example, you may wish to use a mouse over animation), provided that you set the *panning method* parameter to *animated*.



3. Add any actions that you desire to the hotspot movie clips as these behave like regular movie clip buttons. For example:

```
librarySpot.addEventListener(MouseEvent.MOUSE_UP,hotSpotClicked);

function hotSpotClicked(event:MouseEvent) {
  var request:URLRequest = new URLRequest("http://www.flashloaded.com");
  navigateToURL(request, "_blank");
}
```

Defining hotspots through ActionScript

The following sample code creates a hotspot 30px wide by 30px high at the location x: 200 y: 100, on top of the 360PanViewerPRO which has an instance name of *pView*. The name of the hotspot is displayed in the output window when clicking on the hotspot in the Flash IDE:

```
import com.flashloaded.panviewer.events.HotSpotEvent;

pview.addEventListener(HotSpotEvent.CLICK, onClick);
function onClick(e:HotSpotEvent)
{
      trace(e.name);
}


pview.addHotSpot({
      name: "ActionScript HotSpot",
                  area:[200,100,30,30],
                  color:0x00ff00,
                  borderColor:0xffffff,
                  alpha:.5
            });
```

Defining an image through XML

The image that you wish to display in the 360PanViewerPRO-AS3 component, as well as hotspots can be specified using an XML file. Optionally component parameters can also be set in the XML file which will override the same parameters that have been set in the Component Inspector.

By defining the images and hotspots in an external XML file, you can publish the SWF file once and change the content without editing the SWF.

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"?>
<panviewer>
</panviewer>
```

4. Add the source tag to your XML file. This is where you specify the image filename and path (the bold lines are the new additions).

```
<?xml version="1.0" encoding="utf-8"?>
<panviewer>
    <source>images/pano.jpg</source>
</panviewer>
```

You may skip to point #7 if you do not wish to add hotspots or additional parameters to the XML file.

5. You can also set component parameters in the XML file. Any parameter specified will override the setting for the same parameter as set in the Component Inspector (the bold lines are the new additions). The *width* and *height* settings define the size of the view area on the stage.

```
<?xml version="1.0" encoding="utf-8"?>
<panviewer>
    <source>images/pano.jpg</source>
    <method>bitmap</method>
    <width>650</width>
    <height>300</height>
    <mouseAction>press</mouseAction>
    <is360>false</is360>
    <speedIndex>23</speedIndex>
    <slowdownIndex>10</slowdownIndex>
    <maxZoom>2</maxZoom>
    <autopan>
        <horizontalSpeed>4</horizontalSpeed>
        <verticalSpeed>8</verticalSpeed>
        <delay>1</delay>
    </autopan>
    <displacement>
        <angleIndex>60</angleIndex>
        <distortionIndex>120</distortionIndex>
    </displacement>
</panviewer>
```

You may skip to point #7 if you do not wish to add hotspots.

6. Hotspots can also be added to the XML file. You can either define the coordinates of the hotspot area (including the color, border color and alpha), or you can specify the Class name and X/Y coordinates for a movie clip that's in the library that has been set to *export for ActionScript*.

   Adding a hotspot area by plotting the coordinates

   Add the following lines (edit them accordingly) before the closing `</panviewer>` tag. Each hotspot must be defined within its own set of <hs>.....</hs> tags:

```
<hs name="xmlHS">
    <area>500,140,160,85</area>
    <color>0x00ff00</color>
    <borderColor>0xffffff</borderColor>
    <alpha>0.3</alpha>
    <url>http://www.flashloaded.com</url>
    <target>_blank</target>
    <over>over hotspot xmlHS</over>
</hs>
```

Adding a library movie clip as a hotspot

Add the following lines (edit them accordingly) before the closing </panviewer> tag. Each hotspot must be defined within its own set of <hs>.....</hs> tags:

```
<hs name="xmlHS2">
    <id>PV360_HotSpot</id>
    <x>300</x>
    <y>200</y>
    <over>over hotspot xmlHS2</over>
    <load>images/myimage.jpg</load>
</hs>
```

Explanation of hotspot tags

Common tags that be defined for both movie clip and plotted hotspots:

name: Every hotspot must have a unique name.
url (optional): URL to open when clicking on the hotspot.
target (optional): Target window in which to open the URL.
over (optional): Value which can be used through ActionScript in the over event handler. E.g. This can be text which is set through ActionScript to be displayed in a textfield on mouse over.
load (optional): This can be used instead of *url* and *target* to specify an image or SWF file to load when clicking on the hotspot.

Note: Every hotspot should have either target+url or load defined.

Tags for rectangular plotted hotspots only:

area: Coordinates, width and height defining the hotspot rectangle (x,y,width,height). The X/Y coordinates are calculated from the top left corner of the loaded image.
color: HEX Color of the hotspot in the format 0xHEX (e.g: 0xff0000).
borderColor (optional): HEX color of border around hotspot in the format 0xHEX (e.g: 0xff0000).
alpha (optional): Alpha opacity for hotspot (range 0-1). Default value = 0.2.

Tags for movie clip hotspots only:

id: Movie clip Class name of the hotspot (set when choosing *export for ActionScript* for the movie clip).
x: Horizontal position of the top left corner of the hotspot relative to the top left corner of the panned image.
y: Vertical position of the top left corner of the hotspot relative to the top left corner of the panned image.

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

8. Return to your Flash file. Enter the name and path to the XML file that you just created in the ***source*** parameter of the 360PanViewerPRO that's on the stage.

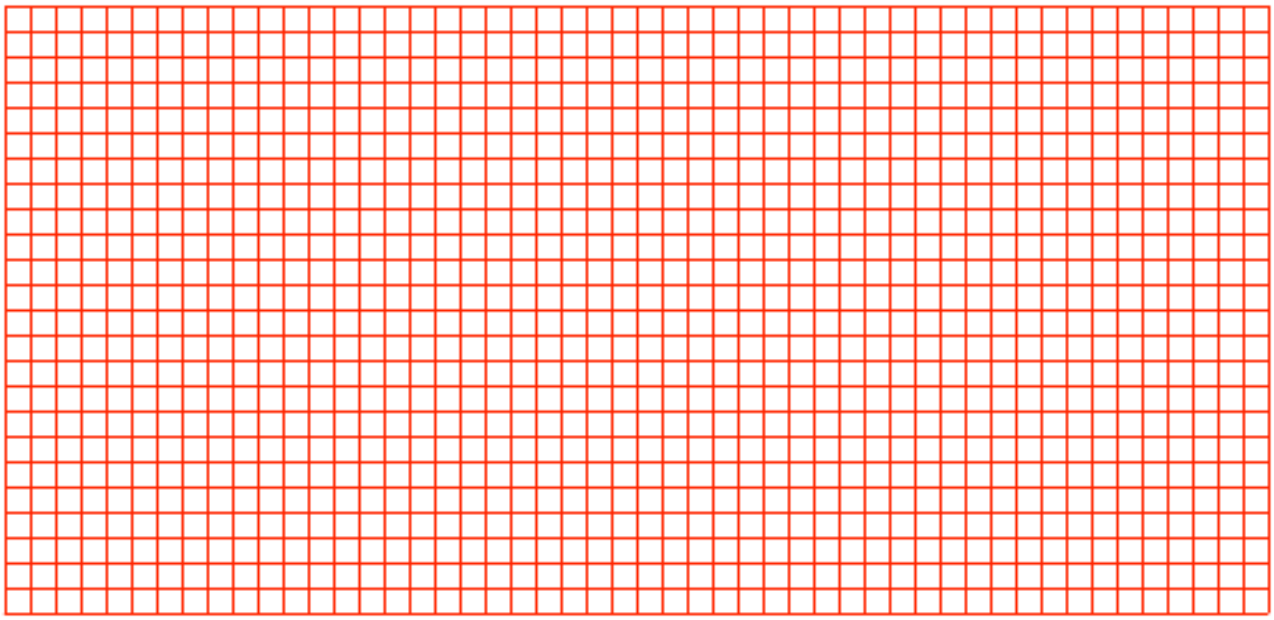
Note: If your .swf file will be in a different folder to the HTML file in which it is embedded, you should enter the path to the XML file, relative to the location of the .html file.

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

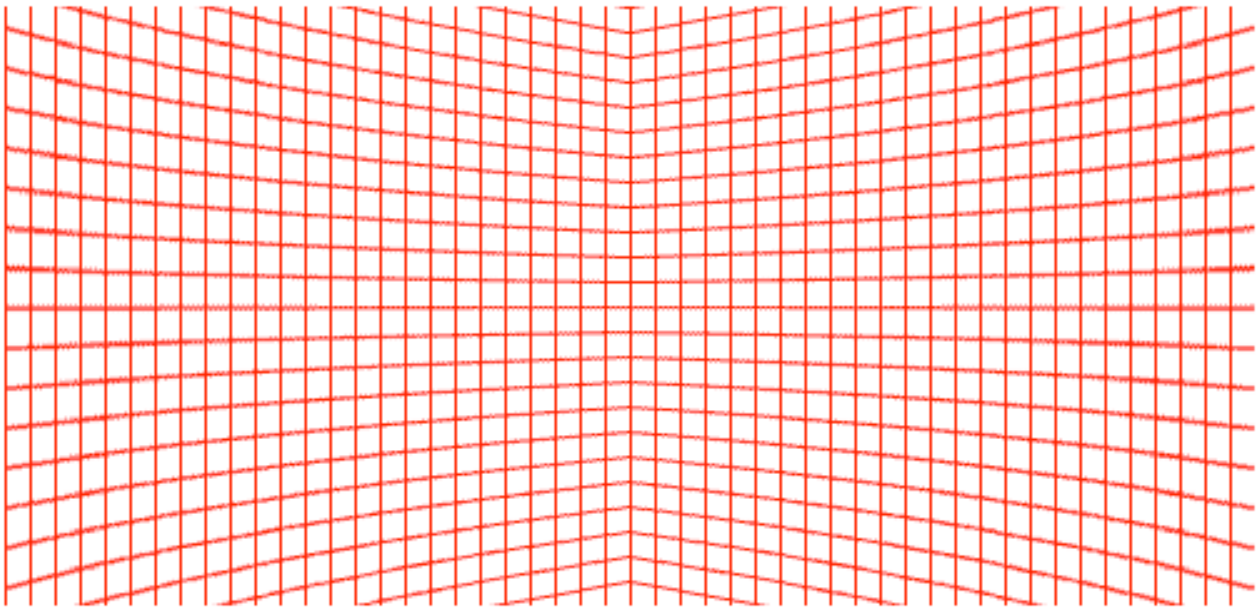
Displacement

Displacement adds depth to your images. There are two parameters that you can set for displacement: angle Index and distortion index. The following images will help you understand how this functions:

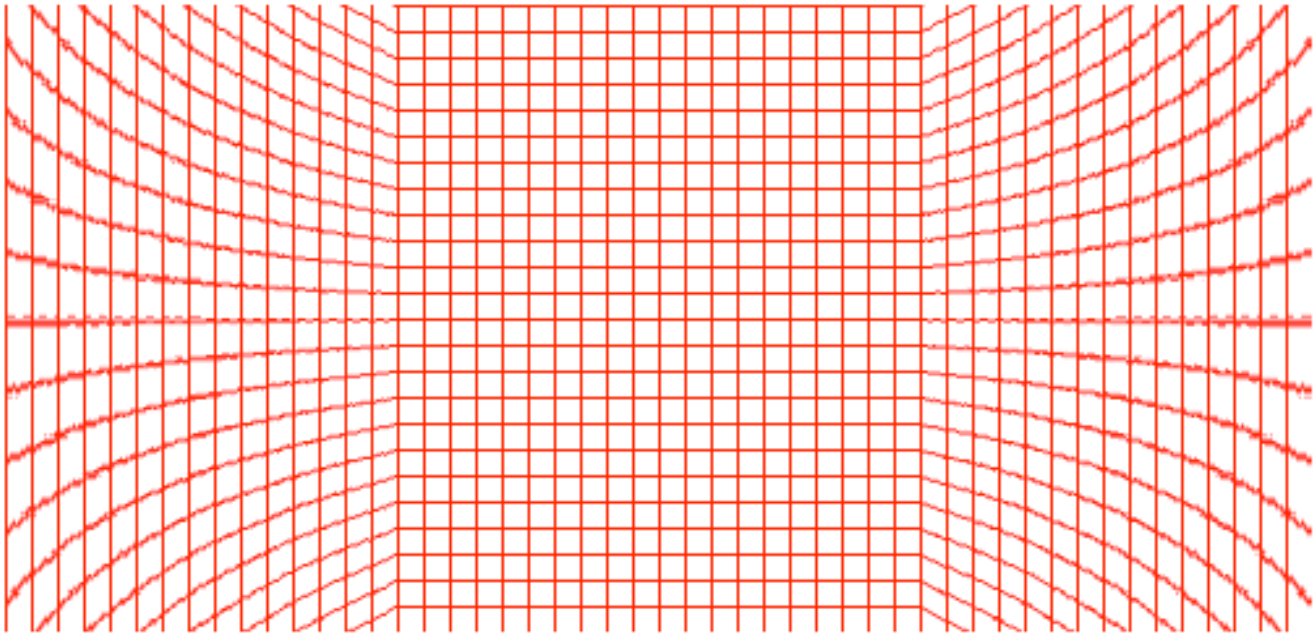
displacement: false



displacement: true
angle index: 0
distortion index: 90

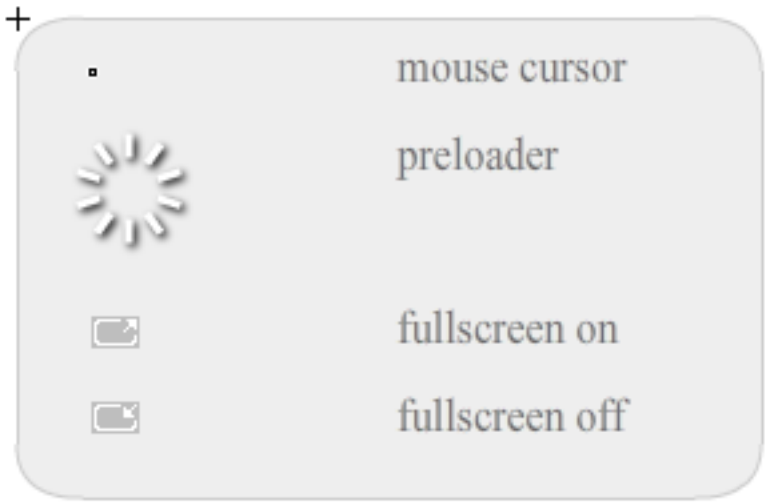


displacement: true
angle index: 40
distortion index: 150



Skinning

All of the graphic elements such as the preloader, mouse cursor and full screen icons, can be skinned. To change the skin, double click on the 360PanViewerPRO component that's on the stage. You can then edit the skin of each element by double clicking on their respective movie clips.



ActionScript methods

Here are some of the ActionScript methods that available. Please click here for the full ActionScript API.

addHotSpot

Availability
Flash Player 9

Description
Method; Defines a hotspot, including the color and opacity.

Example

```
360PanViewerInstance.addHotSpot({
    name:"ActionScript HotSpot",
    area:[200,100,30,30],
    color:0x00ff00,
    borderColor:0xffffff,
    alpha:.5
});
```

LoadXml

Availability
Flash Player 9

Description
Method; sets the XML file to load into the 360PanViewerPRO component.

Example

```
360PanViewerInstance.LoadXML("feeds.xml");
```

source

Availability
Flash Player 9

Description
Method; sets the source of the pan image to use.

Example
```
360PanViewerInstance.source = "image.jpg";
```

zoom

Availability
Flash Player 9

Description
Method; Zooms the image in or out.

Example
```
360PanViewerInstance.zoom = 0.5;
```

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:
360PanViewerPRO-AS3 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.