

PanoramaStudio Viewer 2.0

Panorama Player for Flash and Java

Documentation

© 2005-2008, Tobias Huellmandel Software
<http://www.tshsoft.com>

Contents

1	The PanoramaStudio Viewer	1
1.1	Product info and system requirements	1
1.2	Features	1
1.3	Use with PanoramaStudio	1
1.4	Use with panoramas created by other software	1
1.5	Controlling the Viewer	1
1.6	License	2
2	Manually embed the Viewer into HTML pages	3
2.1	Java version: The <applet> tag	3
2.2	Flash version: The <object> and <embed> tag	3
2.3	Necessary files for the use of the Viewer	4
2.4	Controlling the Viewer with JavaScript	4
3	Description of the parameters	5
3.1	Primary parameters	6
3.2	Auto Play parameters	8
3.3	Toolbar parameters	8
3.4	Hotspot parameters	9
4	Hotspots and virtual tours	9
5	Sample	10

1 The PanoramaStudio Viewer

1.1 Product info and system requirements

The *PanoramaStudio Viewer* serves as a tool for interactive presentations of spherical and cylindrical panoramas. The *PanoramaStudio Viewer* is an Applet which is now available in two versions, as Java based and as Flash based player.

The Java based version requires the Java Runtime Environment (JRE) Version 1.1 or higher. It runs on every Java capable system (Windows, Linux, MacOS,...), if there is enough RAM available for the display of the panoramas.

The Flash based version requires an installed Flash player version 9 or higher. With sufficient system memory for the presentation of the panoramas this version will also run on every compatible system.

Both versions of the Viewer feature the same functionality and interface which allows an easy switching between the two players. The main advantage of the Viewer is the very efficient program code which allows high frame rates and high rendering quality at the same time. Other advantages are the seamless integration in *PanoramaStudio* and the support of hotspots and virtual tours.

1.2 Features

- Interactive and platform-independent presentation of panoramas
- Highly optimized Java and Flash code for high frame rates
- Supports spherical and cylindrical panoramas
- Displays partial panoramas as well as spherical 360x180 degree panoramas
- Controlling the Viewer with JavaScript functions
- Supports hotspots and virtual tours
- Dynamic adjustment of display quality vs. frame rate matching the available CPU power
- Auto-Play for automatic panning, tilting, and zooming in the panorama
- User-interaction by mouse, keyboard, and mouse wheel

1.3 Use with PanoramaStudio

Since version 1.6 the *PanoramaStudio Viewer* is an integral part of the PanoramaStudio installation. Website templates with an embedded panorama can automatically and comfortably be created by PanoramaStudio. PanoramaStudio generates fully automatically the necessary HTML code and the necessary files accordingly to the user's settings for this purpose. This PanoramaStudio export function supports already a large part of the parameters and options of the Viewer described in this documentation.

1.4 Use with panoramas created by other software

As a matter of principle the Viewer accepts and shows also all cylindrical and spherical panoramas created by other software. The condition is that the panoramas are images in JPEG format. This documentation shows in the following chapters how to embed the Viewer along with existing panoramic images into HTML pages.

1.5 Controlling the Viewer

Inside a panorama showed with the *PanoramaStudio Viewer* the user can navigate interactively. The main control device is the mouse. You can pan horizontally and tilt vertically inside the scene by moving the mouse while pressing the left button. Additionally the mouse wheel is supported to zoom in and out, if a Java Runtime Environment Version 1.4 or higher is installed. If the toolbar is shown in the Viewer, there are furthermore buttons for the navigation, for starting and stopping the Auto Play, and for showing/hiding the hotspots, if applicable.

The Viewer can be controlled by keyboard, too. The cursor keys serve here for the navigation as well as the direction keys on the numerical pad. You can zoom in and zoom out by '+' and '-' respectively. The space bar serves to start and stop the Auto Play. With 'h' the existing hotspots can be shown and hidden. The 't' does the same for the toolbar.

1.6 License

License agreement

PanoramaStudio Viewer - Version 2.0
for JavaRE (JRE) Version 1.1 or higher
and for Adobe Flash Player Version 9 or higher
Copyright © 2005-2008, Tobias Huellmandel Software
Internet: <http://www.tshsoft.com>
E-Mail: support@tshsoft.com

1. LICENSE: The author (Tobias Hüllmandel Softwareentwicklung) provides you the Software "PanoramaStudio Viewer 2.0", named "Software" in the following, which contains further an electronic documentation and a license file ("License.txt") and grants you a license for using this product in accordance with the following conditions.
2. COMMERCIAL/PRIVATE USE:
 - 2.1 PRIVATE USE
The private/non-commercial use of the Software, also in the internet, is free and does not require an additional license.
 - 2.2 COMMERCIAL/INDUSTRIAL USE
For every commercial internet presence where the Software is used one (Commercial) License must be purchased.
An internet presence may be defined here by the associated domain (domain in terms of "www.tshsoft.com"). A commercial license is valid for the domain name and possibly existing domain name aliases referring to the same internet presence. The number of panoramas showed with a licensed copy of the Software on an internet presence is not limited.
A commercial use of an internet presence is basically existend, if the owner uses the internet presence with a commercial interest, or if he is no private individual, or if he is no non-profit organisation and no academic institution.
3. LICENSE RESTRICTIONS:
 - 3.1 You may not use this Software in a way other than allowed in this License.
 - 3.2 You shall not use an UNREGISTERED copy of the Software on a commercial internet presence after the end of a 30-day trial period.
 - 3.3 You may not alter, modify, adapt or translate the Software, or decompile, reverse engineer, or disassemble the Software.
 - 3.4 You may not modify the Software or create derivative works based upon the Software.
4. THE SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING ANY WARRANTY OF QUALITY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
IN NO EVENT WILL THE AUTHOR BE LIABLE FOR LOSS OF DATA, COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES OR ANY SPECIAL, CONSEQUENTIAL OR INCIDENTAL DAMAGES, UNDER ANY CAUSE OF ACTION AND REGARDLESS OF WHETHER OR NOT THE AUTHOR HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. THIS LIMITATION WILL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN. IN ANY EVENT THE AUTHOR WILL HAVE LIABILITY ARISING OUT OF THIS AGREEMENT.

Purchase a License

According to the above license agreement a license key is needed for the commercial use of the *PanoramaStudio Viewer* on the internet after a 30-day trial period. For such a commercial use you can buy your license at the registration service Share-It!. With the purchase you will get a license key compatible to the domain name of your internet presence.

With the license key you will get:

- The right to use the *PanoramaStudio Viewer* on a commercial internet presence. The number of panoramas showed with a licensed copy of the Viewer on an internet presence is not limited.
- The possibility to show your own logo in the Viewer window.
- Free help with your questions and issues.

Order a license key

A license key can be ordered easily and primarily secure over the internet at the registration service Share-It!. We accept credit cards, checks, bank/wire transfer, or cash as payment method. The delivery of your license key for a certain domain name is done by e-mail within 24 hours after receiving your payment. If you pay by credit the license key is delivered in most cases within 30 to 60 seconds.

Order per Internet: <http://www.tshsoft.com/en/panostudioapplet/register.html>

Customer service

Do you have questions about your order, payment or the delivery? Do you have already ordered a product and want to check your data? You will find the answers at the Customer Care Center of Share-It!:

<http://shareit1.element5.de/ccc/index.html?publisherid=20959&languageid=1>

2 Manually embed the Viewer into HTML pages

2.1 Java version: The <applet> tag

Embedding the Java version of the *PanoramaStudio Viewer* into a HTML page is done with the <applet> tag. The *PanoramaStudio Viewer Applet* is the file *panoStudioViewer.jar*. The following example shows the most compact type of an <applet> tag which embeds this Java Viewer:

```
<applet archive="panoStudioViewer.jar" code="panoStudioViewer.PanoStudioViewer.class"
  name="MyPanorama" width="800" height="480">
  <param name="pano" value="MyPanorama.xml">
</applet>
```

All necessary parameters to describe and configure the panorama are located in a XML file which is referenced with the parameter `param name='pano'`. In the above example this parameters refers to to the file *MyPanorama.xml*. You find a more detailed description about the structure of the parameter file in chap. 3 (→ p.5).

2.2 Flash version: The <object> and <embed> tag

For the Flash version you have to embed the Flash module of the *PanoramaStudio Viewer* into the webpage. The Flash module is the file *panoStudioViewer.swf*. In order to get the Flash version working with all browsers some more effort is needed than for the Java version. The following example shows the Flash version:

```
<object classid="CLSID:D27CDB6E-AE6D-11cf-96B8-444553540000"
  width="800" height="480" id="MyPanorama"
  codebase="http://active.macromedia.com/flash9/cabs/swflash.cab#version=9,0,28,0">
  <param name="movie" value="panoStudioViewer.swf" />
  <param name="allowScriptAccess" value="always" />
  <param name="allowNetworking" value="all" />
  <param name="allowFullScreen" value="true" />
```

```

<param name="FlashVars" value="pano=MyPanorama.xml" />
<embed src="panoStudioViewer.swf" width="800" height="480"
      type="application/x-shockwave-flash" name="MyPanorama"
      allowScriptAccess="always" allowNetworking="all" allowFullScreen="true"
      FlashVars="pano=MyPanorama.xml"
      pluginspage="http://www.macromedia.com/go/getflashplayer" >
</embed>
</object>

```

In line with the Java version all the parameters for the description of the panorama are located in the XML file *MyPanorama.xml*. But the XML file needs to be referenced here in the **object** as well as in the **embed** tag at the parameter **FlashVars** as shown above.

A great advantage of the *PanoramaStudio Viewers* is that the Flash version as well as the Java version use the same XML parameter file. Thus a panorama can be played by both Viewers without the need to change anything in the XML or the image file.

Furthermore, you have to take care here to set the same parameters **width** and **height** in the **object** and in the **embed** tag. The same is essential for the parameter **id** in the **object** tag and the equivalent parameter **name** in the **embed** tag.

2.3 Necessary files for the use of the Viewer

To publish a website with an interactive panorama, you have to include several files beside the HTML page. This files are basically either the Java applet with the file **panoStudioViewer.jar** or the Flash applet **panoStudioViewer.swf**, and the panorama image in a JPEG image file as well as a XML file which contains all needed information and parameters for the panorama and the Viewer appearance.

For the size of a panoramic image you have to take into account that the Java Viewer as well as the Flash Viewer are subject to certain restrictions and are not able show arbitrary sized images. For the Java version the image size is limited by the memory which is available to the Java plugin. Therefore the maximum size of the panoramic image shouldn't exceed 15 or 20 megapixels. For Flash most versions of the Flash Player limit the maximum width and height of an image to 8192 pixels. Loading larger images is not possible.

If there are hotspots used a hotspot file must be included, too. Please read chap. 4 (→ p.9) for this topic. All files should be located in the same directory, if possible. Otherwise you have to set appropriate file paths in the HTML code and in the XML file.

2.4 Controlling the Viewer with JavaScript

Both versions of the *PanoramaStudio Viewer* can be controlled by other components from the website with JavaScript commands.

Notice: For security reasons the Flash Player does not allow controlling a Flash applet with JavaScript located on the local computer. Hence the JavaScript commands will only work if the *PanoramaStudio Viewer* runs online on a webserver.

The following JavaScript commands are available for the Viewer:

- **void startAutoRotate()**
Starts the Auto Play.
- **void stopAutoRotate()**
Stops the Auto Play.
- **void toggleHotspots()**
Shows/hides existing hotspots.
- **void setAutoRotate(double autoPanRate, double autoTiltRate, double autoZoomRate)**
Sets the parameters for the Auto Play. The meanings of the arguments **autoPanRate**, **autoTiltRate**, and **autoZoomRate** are here equivalent to the parameters of the same name in chap. 3.2 (→ p.8).
- **void showToolbar()**
Shows the toolbar in the viewer window.

- `void hideToolbar()`
Hides the toolbar.
- `void setView(double pan,double tilt,double hfov)`
Changes the viewing direction in the panorama to the angles `pan` and `tilt` as well as the field-of-view to the value `hfov`.
- `boolean isRotating()`
Checks, if the Auto Play is active.
- `void openPanorama(String filename)`
Opens another panorama in the same Viewer window. For that purpose the parameter `filename` must contain the name of a XML parameter file.

JavaScript samples

Here are two examples of a start and a stop button for the Auto Play function. `MyPanorama` is in each case the name of the `<applet>` object (as shown in example chap. 2.1 (→ p.3)).

```
<INPUT type="button" value="Play" onClick="document.MyPanorama.startAutoRotate();">
<INPUT type="button" value="Stop" onClick="document.MyPanorama.stopAutoRotate();">
```

A more complex example of a combined play/pause button.

```
<INPUT type="button" value="Play/Pause" onClick="
  if (document.MyPanorama.isRotating())
    document.MyPanorama.stopAutoRotate();
  else document.MyPanorama.startAutoRotate();
">
```

Here is an example of a JavaScript action on an `<a href>` tag. The Auto Play direction will be changed here:

```
<a href=# onClick="document.MyPanorama.setAutoRotate(-2.0,-1.0,1.0);">Change direction</a>
```

3 Description of the parameters

All parameters regarding the properties of the panorama and the appearance of the Viewer are contained in a XML file:

The structure of this XML for the description of the panorama properties is composed of the XML declaration, the `<panoramaStudioViewer>` element and a `<panorama>` element which is enclosed therein. The XML declaration contains the used XML version and character encoding. This has to be always as follows for the *PanoramaStudio Viewer*:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

This results in the following basic structure for a XML description file:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<panoramaStudioViewer>
  <panorama>
    [...]
  </panorama>
</panoramaStudioViewer>
```

All necessary parameters are listed within the `<panorama>` element. There, every parameter is again a XML element composed of an opening and closing tag. An example for the `file` parameter:

```
<file>Panorama.jpg</file>
```

In detail the Viewer knows the following parameters:

3.1 Primary parameters

- **file**
Filename/URL
Filename of the panoramic image. The image must be a file in JPEG format.
- **hsfile**
Dateiname/URL
Filename of an optional hotspot image. The image must be a file in GIF format and a color depth of 8 bit. See also chap. 4 (→ p.9).
- **key**
String
License key parameter. This parameter is necessary for the use on commercial websites. Please read the license agreement for more information.
- **lang**
String
Language setting for the Viewer. Available is **en** for english, **fr** for french, and **de** for german.
- **pan**
Float
Initial horizontal viewing direction (pan).
- **tilt**
Float
Initial vertical viewing direction (tilt).
- **hfov**
Float
Initial horizontal field-of-view.
- **minpan**
Float; default: 0.0
Minimum pan. **minpan** limits the horizontal pan to the left.
- **maxpan**
Float; default: 360.0
Maximum pan. **maxpan** limits the horizontal pan to the right.
- **mintilt**
Float
Minimum tilt. **mintilt** limits the vertical tilt value on the upper border.
- **maxtilt**
Float
Maximum tilt. **maxtilt** limits the vertical tilt value on the lower border.
- **minhfov**
Float; [5.0;165.0]; default: 5.0
Minimum horizontal field-of-view. The ability to zoom into the image is limited by this value.
- **maxhfov**
Float; [5.0;165.0]; default: 165.0
Maximum horizontal field-of-view. The ability to zoom out of the scene is limited by this value.
- **spherical**
Boolean; 'true'/'false'; default: 'true'
Declares the source image as spherical or cylindrical panorama.
- **quality**
Integer; [0;4]; default: 1
Setting of the display quality.
Levels: 0: lowest/fastest, 1: low, 2: medium, 3: hi quality/slowest frame rate, 4: dynamic

At high quality the rendered image will be interpolated bilinearly. At level 0 the interpolation is always disabled. At higher levels the interpolation will be activated step by step in the phases of displaying still image, Auto Play, and user interaction. At Level 4 (dynamic) the image quality will be set automatically in dependence of the possible frame rates on the current computer.

- **showToolBar**
Boolean; 'true'/'false'; default: 'true'
Shows/Hides the toolbar.
- **waitImage**
Filename/URL
Optional image which is displayed in the waiting screen while loading the panorama scene. The image has to be a file in PNG, JPEG, or GIF format.
- **logo**
Filename/URL
Filename or URL of a custom logo which is showed instead of the PanoramaStudio logo. This function is only available if the applet runs on the local computer or if a valid license key is assigned to the **key** parameter, when it runs online.
- **logoUrl**
Filename/URL
URL which is assigned to the custom logo.
- **logoX** and **logoY**
Integer; default: 4
Position of the custom logo relative to the upper-right corner.
- **bgWait**
RGB color; Hexadecimal; default: 000000
Background color of the waiting screen while loading the panorama.
- **fgWait**
RGB color; Hexadecimal; default: FFFFFFFF
Foreground color of the waiting screen while loading the panorama.
- **loadString**
String; default: 'Loading:'
Waiting text displayed in front of the progress bar's percentage indication.
- **title**
String
Title of the panorama. Will be shown in the status line.
- **showBorder**
Boolean; 'true'/'false'
Draws a border line around the applet window.
- **borderColor**
RGB color; Hexadecimal; default: 000000
Border color of the applet window.
- **textColor**
RGB color; Hexadecimal; default: FFFFFFFF
Text color of the status text in the applet window.
- **shadowColor**
RGB color; Hexadecimal; default: 000000
Shadow color which underlays the status texts.
- **showHotspots**
Boolean; 'true'/'false'
Shows or hides the hotspots.

- **mass**
Float; [0.1;2.0]; default: 0.5
Mouse mass. The mouse mass assigns an inertia to the movements in the scene caused by the user's mouse actions. So, there are no abrupt changes of the moving direction but the movements are rather slowing down an speeding up according to the mouse mass.
- **mouseSen**
Float; [0.1;2.0]; default: 1.2
An amount and speed of movement in the scene caused by the user's mouse actions respectively is defined by this value.

3.2 Auto Play parameters

Parameters regarding the auto play are expected within a <autoplay> element:

- **autoRotate**
Boolean; 'true'/'false'; default: 'false'
Enables the Auto Play.
- **autoPanRate**
Float; default: 2.0
Panning speed for the Auto Play (in degrees/sec).
- **autoTiltRate**
Float; default: 0.0
Tilting speed for the Auto Play (in degrees/sec).
- **autoZoomRate**
Float; default: 1.0
The Auto Play zoom. At a value of 1.0 the zoom and the field-of-view respectively remains constant. It zooms in at values less than 1.0, and zooms out at values greater than 1.0.
- **autoRotateRestart**
Integer; default: 5
Time until the Auto Play restarts after an interruption caused by user interaction.

3.3 Toolbar parameters

Parameters regarding the toolbar are expected within a <toolbar> element:

- **tbLayout**
Integer; [0;2]; default: 0
Here you can choose one out of three predefined toolbar layouts.
- **tbPosition**
Integer; [0;1]; default: 0
Position of the toolbar within the Viewer window (0: bottom,1: top)
- **tbButtonType**
Integer; [0;1]; default: 0
Type of the toolbar buttons. Type 0 means buttons with rounded edges, Type 1 means buttons with square edges.
- **tbColorProfile**
Integer; [0;4]; default: 0
Here you can choose one out of five predefined color profiles for the toolbar.
- **tbShowBackground**
Boolean; 'true'/'false'
Underlays the toolbar buttons with a background.

- **tbBGOpacity**
Integer; [0;255]; default: 128
 Opacity of the toolbar background. The background is completely opaque for the setting 255 and invisible at 0.
- **tbButtonOpacity**
Integer; [0;255]; default: 192
 Opacity of the toolbar buttons.
- **tbHotButtonOpacity**
Integer; [0;255]; default: 255
 Opacity of the active toolbar buttons. A button is active if the mouse cursor is hovering over the button.
- **tbShowPlayButton**
Boolean; 'true'/'false'
 Shows a 'Play' button in the toolbar which function is to start and stop the Auto Play.
- **tbButtonWidth**
Integer; [18;48]; default: 24
 Width of the toolbar buttons.
- **tbButtonHeight**
Integer; [18;48]; default: 20
 Height of the toolbar buttons.

3.4 Hotspot parameters

There can be up to 255 hotspots per panorama. Hotspots are defined as **hotspotN** XML elements where N has to be in the range of [0;254]. Within a **hotspotN** element the following elements are used for the description of this hotspot:

- **url**
String
 URL to open if the hotspot is clicked.
- **target**
String
 The HTML target for this hotspot. The HTML target is, as specified in the HTML standard, the identifier for the destination frame or a new window ('_blank') where the referred URL should be opened.
 A special case in the *PanoramaStudio Viewer* is the keyword 'myself'. Using this keyword causes the referred panorama to open within the same Viewer window. Therefore the corresponding URL has to refer to the XML file of this panorama.
- **comment**
String
 Comment for this hotspot. The comment is shown in the status line of the Viewer if the mouse hovers above the hotspot.
- **hscolor**
RGB color; Hexadecimal
 Color for this hotspot in hexadecimal format.

4 Hotspots and virtual tours

As described in the parameters chapter (chap. 3.1 (→ p.6)), the *PanoramaStudio Viewer* supports hotspots. Hotspots represent links to other panoramas or websites like common HTML hyperlinks. A hotspot is a certain area in the panorama where on a click on this area opens the linked object. So called *virtual tours* can be created by connecting several panoramas in this manner. This gives you the ability to create e.g. a virtual tour through all the rooms of a museum.

Position and shape of the hotspots are stored since version 2.0 of the *PanoramaStudio Viewer* in an image file in GIF format with 8 bit color depth. Panoramic image and hotspot image must be exactly the same size. Every hotspot is defined by an individual color in this image file. Taking the background into account this limits number of possible hotspots to 255 per panorama.

The color index 0 represents the background in the hotspot image. Then, the first hotspot has the color index 1, and so on. The URL link and other properties of the first hotspot are set by the parameters within the element `hotspot0` with `url`, `target`, `hscolor`, and `comment` (see also chap. 3.4 (→ p.9)). More hotspots are added in the same manner.

Within the XML parameter file the hotspot image file is added to the panorama with the `hsfile` parameter.

5 Sample

The following example shows the XML parameter file of a panorama. As panoramic image serves the file 'panorama_1.jpg' and as hotspot image file the file 'panorama_1.gif'. As defined in the parameters section the panorama has a horizontal field-of-view of 236,51° degrees and the image is a spherical projection of the panorama. The panorama has a hotspot (`hotspot0`) which refers to the file 'panorama_2.xml'. This hotspot opens due to the 'myself' keyword and the reference to a XML parameter file the linked panorama within the same Viewer window.

```
<?xml version="1.0" encoding="UTF-8" ?>
<panoramaStudioViewer>
  <panorama>
    <file>panorama_1.jpg</file>
    <hsfile>panorama_1.gif</hsfile>
    <mintilt>-41.53</mintilt>
    <maxtilt>41.53</maxtilt>
    <minpan>0.00</minpan>
    <maxpan>236.51</maxpan>
    <pan>118.26</pan>
    <tilt>0.00</tilt>
    <hfov>70.00</hfov>
    <spherical>false</spherical>
    <mouseSen>1.00</mouseSen>
    <loadString>Loading:</loadString>
    <title></title>
    <quality>4</quality>
    <autoplay>
      <autoPanRate>3.0</autoPanRate>
    </autoplay>
    <toolbar>
      <showToolbar>true</showToolbar>
    </toolbar>
    <hotspot0>
      <url>panorama_2.xml</url>
      <target>myself</target>
    </hotspot0>
  </panorama>
</panoramaStudioViewer>
```

Within HTML the following code will embed this panorama 'panorama_1.xml' as Java applet:

```
<applet archive="panoStudioViewer.jar" code="panoStudioViewer.PanoStudioViewer.class"
  name="pano" width="600" height="400">\\
  <param name="pano" value="panorama_1.xml">\\
</applet>\\
```

Embedding the same panorama in the Flash version would look as follows:

```
<object classid="CLSID:D27CDB6E-AE6D-11cf-96B8-444553540000"
    width="600" height="400" id="pano"
    codebase="http://active.macromedia.com/flash9/cabs/swflash.cab#version=9,0,28,0">
    <param name="movie" value="panoStudioViewer.swf" />
    <param name="allowScriptAccess" value="always" />
    <param name="allowNetworking" value="all" />
    <param name="allowFullScreen" value="true" />
    <param name="FlashVars" value="pano=panorama_1.xml" />
    <embed src="panoStudioViewer.swf" width="600" height="400"
        type="application/x-shockwave-flash" name="pano"
        allowScriptAccess="always" allowNetworking="all" allowFullScreen="true"
        FlashVars="pano=panorama_1.xml"
        pluginspage="http://www.macromedia.com/go/getflashplayer" >
    </embed>
</object>
```