

# Ryubin's Flash Panorama Player User's Guide

---

Version 5.0

Release 1.0 (November 6, 2011)

Ryubin's Panorama Laboratory

This document is for informational purposes only. Ryubin Office makes no warranties, express, implied, or statutory as to the information in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice.

## Table of Contents

<b>Introduction.....</b>	<b>- 6 -</b>
<b>One-of-a-Kind Benefit.....</b>	<b>- 8 -</b>
<b>Pros and Cons.....</b>	<b>- 10 -</b>
<b>Playmode.....</b>	<b>- 11 -</b>
<b>User Interface .....</b>	<b>- 18 -</b>
<b>Simple approach using built-in controller .....</b>	<b>- 20 -</b>
Basic setup .....	- 20 -
<b>Sophisticated approach using XML files .....</b>	<b>- 22 -</b>
Basic setup .....	- 22 -
<b>Typical usages of major XML tags, elements and attributes .....</b>	<b>- 25 -</b>
Specifying panorama image, video, sound to play .....	- 25 -
User definabe mouse cursor.....	- 26 -
Initial viewing experience .....	- 26 -
Panorama controller .....	- 27 -
Sound controller (still only) .....	- 28 -
Video controller (video only) .....	- 29 -
User definable panels (or images).....	- 30 -
Defining panowindow .....	- 30 -
User definable skin.....	- 32 -
User definable logo image .....	- 33 -
User definable system messages.....	- 33 -
Customizing the auto panning behaviors .....	- 34 -
Managing auto panning operation.....	- 34 -
Enabling sinusoidal panning .....	- 34 -
<b>Customizing the image projection .....</b>	<b>- 35 -</b>
Cropping the image .....	- 35 -
Projecting the image.....	- 37 -
Handling partial-equirectangular image .....	- 40 -
Specifying "hv_ratio" for Flat (2D) playmodes.....	- 41 -

Limiting vertical viewing-direction .....	41 -
Limiting horizontal viewing-direction .....	42 -
Flipping the image.....	42 -
<b>Hotspots.....</b>	<b>43 -</b>
<b>Customizing the context menu.....</b>	<b>44 -</b>
Adding an item to reset viewing experience (home position) .....	44 -
Adding an item to enable/disable smoothing filter .....	44 -
Adding an item to enter/exit full-screen mode .....	45 -
Adding an item to start/stop auto panning .....	45 -
Adding an item for "sphere context" .....	45 -
Adding an item for "fisheye context" .....	46 -
Adding an item to show/hide system information (viewing experience) .....	47 -
<b>Customizing the loading indicator (still playmode only) .....</b>	<b>47 -</b>
<b>Balancing the quality and performance (video only) .....</b>	<b>48 -</b>
<b>Configuring video starting and preloading behavior (video only) .....</b>	<b>50 -</b>
<b>Allowing cross domain access .....</b>	<b>53 -</b>
<b>Allowing the use of JavaScript API.....</b>	<b>54 -</b>
<b>JavaScript API .....</b>	<b>55 -</b>
<b>Publishing .flv files on the Web server (video).....</b>	<b>59 -</b>
<b>Specifications of FlashVars (HTML) parameters .....</b>	<b>60 -</b>
<b>Specifications of the primary XML file.....</b>	<b>62 -</b>
<b>&lt;stage&gt; tag.....</b>	<b>62 -</b>
<panowindow /> element .....	62 -
<cursor /> element.....	62 -
<sandglass /> element .....	63 -
<panorama_controller /> element .....	63 -
<sound_controller /> element .....	64 -
<video_controller /> element.....	64 -
<captions /> element .....	64 -
<controller_font /> element .....	65 -
<security /> element .....	66 -
<b>&lt;system_msgs&gt; tag.....</b>	<b>67 -</b>
<center_msg /> element .....	67 -
<hotspots_enabled /> element .....	67 -

<b>&lt;play_objects&gt; tag .....</b>	<b>- 68 -</b>
<image /> element.....	- 68 -
<sound /> element .....	- 68 -
<crop /> element .....	- 69 -
<projection /> element.....	- 69 -
<view /> element.....	- 70 -
<auto /> element .....	- 72 -
<behavior /> element.....	- 72 -
<info /> element .....	- 73 -
<caps /> element .....	- 74 -
<b>&lt;hot_spots&gt; tag .....</b>	<b>- 74 -</b>
<add /> element.....	- 74 -
<b>&lt;play_list&gt; tag .....</b>	<b>- 75 -</b>
<add /> element.....	- 75 -
<b>&lt;user_loader&gt; tag.....</b>	<b>- 75 -</b>
<caption /> element.....	- 75 -
<indicator /> element .....	- 76 -
<background /> element .....	- 76 -
<b>&lt;user_panel&gt; tag .....</b>	<b>- 76 -</b>
<prologue /> element .....	- 76 -
<u_context1 /> element .....	- 77 -
<u_context2 /> element .....	- 77 -
<skin /> element.....	- 77 -
<logo /> element .....	- 78 -
<b>&lt;context_menu&gt; tag.....</b>	<b>- 78 -</b>
<u_context1 /> element .....	- 78 -
<u_context2 /> element .....	- 79 -
<home_pos /> element .....	- 79 -
<show_info /> element.....	- 79 -
<smoothing /> element.....	- 80 -
<panorama_controller /> element .....	- 80 -
<hot_spots /> element .....	- 80 -
<sound_controller /> element .....	- 81 -
<video_controller /> element.....	- 81 -
<start_auto /> element.....	- 81 -
<full_screen /> element .....	- 82 -

<about_me /> element .....	- 82 -
<b>&lt;sphere_context&gt; tag .....</b>	<b>- 82 -</b>
<normal /> element .....	- 82 -
<globe /> element .....	- 83 -
<bipolar /> element .....	- 83 -
<b>&lt;fisheye_context&gt; tag .....</b>	<b>- 83 -</b>
<single /> element .....	- 83 -
<double /> element .....	- 84 -
<mirror /> element .....	- 84 -
<b>Specifications of the XML-based customizable controller .....</b>	<b>- 85 -</b>
<b>Format .....</b>	<b>- 85 -</b>
<b>General rule .....</b>	<b>- 85 -</b>
<attribute table> .....	- 86 -
<action table> used for button objects .....	- 87 -
<role table-A> used for image objects .....	- 88 -
<role table-B> used for text objects .....	- 89 -
<b>Basic design concept .....</b>	<b>- 89 -</b>
Sample code 1 (IntegratedCtrl.xml) .....	- 90 -
Sample code 2 (ArrowButtons.xml) .....	- 92 -

## Introduction

Thank you for using Ryubin's Flash Panorama Player (RFP player). We do hope you enjoy using this state of the art technology made available via Adobe® Flash™ which is the most pervasive multimedia platform runs on any operating system including Windows, Mac, Linux and Android. Thanks to the innovation of information technology built on a range of critical elements including semiconductor chips, the always-connected Internet infrastructure, the Cloud application deployment, and more importantly the increasingly advanced IDE (Integrated Development Environment), even an individual or a small group of people like us are now able to unleash their imagination once they have sufficient enthusiasm towards making their dreams come true, regardless of their financial condition.

Operated by Ryubin Office, Ryubin's Panorama Laboratory, a small technology house who develops and releases the RFP player, is focused on investigating the possibility of personal computing environment in terms of presenting ultra wide angle still images as well as motion pictures without implementing any specific or dedicate hardware or devices, with a hope of realizing an affordable approach or solution allowing global panorama fans and hobbyists to enjoy and share the world of omni-directional vision with minimum additional investment.

RFP player is provided as a small SWF file executable on the Flash platform, meaning that an appropriate version of Adobe's Flash Player is required to be installed in the users computing environment so that RFP player is able to properly function. We do recommend to visit Flash download page (<http://get.adobe.com/flashplayer/>) to obtain the latest version of the player while the prerequisite for current version of RFP player is Flash Player 10 or higher.

Readers of this document are expected to have minimum technical knowledge and skill to run SWF files on their local computing platform or to transfer SWF files along with media files as well as associated files onto your Web server to make them visible to the global Internet audiences. The skill includes small amount of knowledge about HTML, JavaScript, media file managements and FTP (File Transfer Program) or equivalent mean to upload necessary files onto the Web server. If you are familiar with XML, RFP

player will give you a range of flexibility to make the most of their capability while the RFP player is operable without using the XML at their default setups.

Our approach is to develop and deliver a set of primitive components and templates allowing the users to freely put them together and deploy their own setups or combinations, instead of creating some sophisticated or higher level authoring tools to streamline the process of publishing omni-directional images and motion pictures. Our position is that we do welcome some other individual, team, group, or whatever entity to start the development of such authoring tools or platforms leveraging the RFP player as part of their deployments or solutions.

Please note that the contents of this document are focused on conveying basic information for the readers to handle the RFP player, and are not designed to address the approaches or how-to of capturing panoramic pictures nor shooting panoramic motion pictures.

In addition, we would like dear readers to understand that this document is for informational purposes only. We are not able to make any warranties, express, implied, or statutory as to the information in this document. Besides, the information contained in this document, including URL as well as other Internet Web site references, are subject to change without any notice.

Thank you,

Ryubin's Panorama Laboratory  
Ryubin Office, Kanagawa Japan

## One-of-a-Kind Benefit

RFP player has a dozen of breaking-through features, providing one-of-a-kind benefit to the users:

- ◆ Freeware (which is distributed for free of charge for any purpose).
- ◆ Supports multiple platforms; Windows, Mac, Linux and Android (Flash intrinsic).
- ◆ Supports multiple still-image formats such as jpg, png and gif (Flash intrinsic).
- ◆ Supports video-file formats such as flv and f4v (Flash intrinsic). Visit [Adobe Technology Center](#) for details.
- ◆ Supports multiple types of spherical panorama image/video:
  - full ER<sup>(\*)</sup> (sphere, spherevideo playmodes)
  - 8-partitioned ER (sphere8 playmode)
- ◆ Supports multiple types of cubic panorama image/video:
  - normal 6 faces (cube6 playmode)
  - single image containing 6 faces (cube2x3, cube3x2, cube2x3video, cube3x2video)
  - single image containing 4 faces (4sidedvideo)
  - 4-partitioned 6 faces (cube24)
- ◆ Provides a standalone (off-line) tool “PanoSphereConv” to generate images for cube, cube6, cube24, sphere and sphere8, from an ER image (Windows only).
- ◆ Supports unparalleled number of projection conversions for still images and motion pictures:
  - ER to rectilinear (i.e. rectilinear, normal or perspective)
  - ER to orthogonal (i.e. orthographic or globe)
  - ER to bipolar (i.e. hyperbolic or full-stereographic)
  - Fisheye (hemisphere equidistance) and dual-fisheye to rectilinear
  - Fisheye and dual-fisheye to orthogonal
  - Fisheye and dual-fisheye to bipolar
  - Fisheye to ER (horizontal and vertical)
  - Cube to rectilinear
  - Cylinder to rectilinear
  - Doughnut to rectilinear
  - Doughnut to orthogonal



- Doughnut to bipolar
- Doughnut to ER (video only)
- ◆ Supports hotspots.
- ◆ Supports full-screen mode.
- ◆ Supports XML-based parameter control.
- ◆ Supports multiple interfaces for controlling panorama experiences such as pitch, yaw, FOV etc., and for managing video experiences:
  - mouse (right button, left button, and wheel)
  - keyboard
  - context menu
  - panorama controller (built-in, SWF-based and XML-based external )
  - video controller (built-in, SWF-based and XML-based external )
- ◆ Supports mp3 sound.
- ◆ Supports customizable context menu (i.e. changeable default):
  - customizable menu item
  - customizable caption for menu item
- ◆ Supports user-definable image-loading progress-indicator (still playmodes only).
- ◆ Supports user-definable sprite objects (jpg, png, gif and SWF files can be shown over the panorama image).
- ◆ Supports user-definable skin.
- ◆ Supports "panorama sound" which rotates in sync with the yaw value.
- ◆ Supports typamatic (\*\*) buttons for controlling yaw, pitch and fov.
- ◆ Supports on-the-fly window resizing.
- ◆ Supports implicit retrieving of play objects as well as system components.
- ◆ Supports customizable system messages.

(\*) ER stands for equirectangular.

(\*\*) A typamatic button repeatedly operates while being pressed until released whereas a non-typamatic button operates only when pressed.

## Pros and Cons

Because of its pervasiveness and easy-to-use nature, the Adobe® Flash™ platform is now overwhelming other cross-platform media environments such as Java, and is said that Flash is now installed in more than 90% of global personal computing environments, evidencing the Flash is now the most pervasive cross-operating-system multi-media platform for all of the application workloads from the world of hobby, gaming, Web based video entertainment to more critical business scenarios.

In order to make this breaking pervasiveness happen, the Adobe Flash platform implements an abstraction layer to absorb the differences among operating systems; Windows, Mac, Linux and Android. In turn, this layer gives you a burden when manipulating bits and bytes of the images, and this becomes an intrinsic disadvantage of this sort of platform when handling panorama images especially when trying to work on the projection conversions for motion pictures.

Practically, especially when working on panorama video, panorama movie, omni-video, MotionVR or whatever, users might be concerned about a kind of trade-off among screen size, stage size, video size (width x height), FPS (frame per second), encoding schemes, bitrates and others, and might need to determine an appropriate combination among those factors for your own working environment or designated client environment through an experimental process. Therefore, the truth is that selecting Flash platform might not be an ideal approach to execute such a heavy bitmap operation as in the coordinate transformations of motion pictures, especially for people preferring quality to pervasiveness.

Even that, we are optimistic about the future of these Flash based panorama players or viewers, not only because Adobe is already making great efforts to ease the weakness on bitmap management and also for 3D coordinate transformation, but thanks to the ever-innovating semiconductor technology which is still following the rule of Moore's Law, the chip speed of personal computing devices and environments will become promisingly faster than ever. Flash-based panorama viewers will go onward and upward over time.

## Playmode

The playmode of RFP player represents the way of re-mapping or transforming the target bitmap of the designated still image or the motion picture into a meaningful type of viewing experience such as the normal perspective or rectilinear, the stereographic and other interested way of projections. The RFP player, as a single Flash-based application, is capable of running in multiple play modes by specifying the play mode via an HTML parameter (or FlahVars), in such way like "playmode=spherevideo".

Available playmodes are:





Available playmodes	
For motion pictures	4sidedvideo, bloggiewideo, bloggiewideoFV, bloggiewideo2, cube3x2video, cube2x3video, cylindervideo, doughnut2rectvideo, doughnutvideo, dualfisheyevideo, fish2rectvideo, fisheyevideo, flatvideo, simpleflatvideo, spherevideo, verticalfish2rectvideo, verticalfisheyevideo
For still images	cube3x2, cube2x3, cube6, cube24, cylinder, doughnut, dualfisheye, fish2rect, fisheye, flat, simpleflat, sphere, sphere8, verticalfish2rect, verticalfisheye

Please visit <http://www.ryubinpanorama.com/download.html> to grasp how each of the play modes of the RFP player accommodates a range of approaches to realize its dynamic viewing experiences.




Here is a sample HTML code to show where to specify the playmode.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>RyubinPanoPlayer</title>
<script type="text/javascript">AC_FL_RunContent = 0;</script>
<script src="js/AC_RunActiveContent.js" type="text/javascript"></script>
</head>
<body bgcolor="#333333" >
<script type="text/javascript">
AC_FL_RunContent(
'codebase', 'http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=10,0,0,0',
'pluginspage', 'http://www.macromedia.com/go/getflashplayer',
'bgcolor', '#333333',
'width', '640',
'height', '480',
'id', 'RyubinPanorama',
'movie', 'swf/RyubinPanoPlayer5',
'FlashVars', 'playmode=sphere&internal_ctrl=no&xml_path=xml/sphere_CC.xml',
'allowFullScreen', true,
'allowScriptAccess', 'always'
); //end AC code
</script>
</body>
</html>
```

The following table will help you understand the features of major playmodes implemented in the RFP player:




Playmode	Key functions
sphere (S in short)	<p>Reads an equirectangular type of still image (jpg, png or gif format) (*1) , and dynamically generates a normal (i.e. rectilinear or perspective) (*2) panoramic view, as well as orthogonal (i.e. globe or orthographic) (*3) and stereographic (i.e. bipolar or full-stereographic) (*4) views.</p>  <p>(*1)</p>  <p>(*2)</p>  <p>(*3)</p>  <p>(*4)</p>
spherevideo (SV)	<p>Reads an equirectangular type of video image (flv or f4v format), and dynamically generates a normal view, as well as orthogonal and stereographic views. Namely the video version of the sphere playmode.</p>





## Ryubin's Flash Panorama Player, User's Guide

sphere8 (S8)	<p>Reads 8 split images created from an equirectangular type of still image (jpg, png or gif format) (*5) , and dynamically generates a normal view, as well as orthogonal and stereographic views.</p>  <p>(*5)</p>
cube2x3 (C2x3)	<p>Reads a single still image containing images for 6 faces of a texture cube (*6), and dynamically generates a normal view.</p>  <p>(*6)</p>
cube3x2video (CV3x2)	<p>Reads a video clip whose frame image contains a single image for 6 faces that can compose a texture cube (*6a), and dynamically generates a normal view.</p> 






## Ryubin's Flash Panorama Player, User's Guide




	(*6a)
cube6 (C6)	<p>Reads 6 separate images (*7) for 6 faces of a texture cube, and dynamically generates a normal view.</p>  <p>(*7)</p>
cube24 (C24)	<p>Reads 24 images (*8) split from the images for 6 faces of a texture cube, and dynamically generates a normal view.</p>  <p>(*8)</p>
flat (FL)	<p>Reads an equirectangular type (or whatever type) of still image (*1), and dynamically generates an expanded equirectangular view (*9).</p>  <p>(*9)</p> <p>RFP player in the flat playmode actually reads any type of still image, serializes it, and dynamically generates a horizontally concatenated view of the image.</p>
flatvideo (FLV)	Video version of the flat playmode.
fisheye (F)	<p>Reads a full-circular fisheye image (assumed as equidistance projection) (*10), and dynamically generates a normal view as well as orthogonal and stereographic views. While a full circular fisheye image usually covers only a half of the space (hemisphere), RFP player in the fisheye playmode is capable of internally duplicating the image serial-wise (*11) or mirror-wise (*12) and dynamically generates a pseudo full-space view.</p>

	 <p>(*10)</p>  <p>(*11)</p>  <p>(*12)</p>
fisheyevideo (FV)	Video version of fisheye playmode.
dualfisheye (DF)	Reads two full-circular fisheye images (assumed as equidistance projection) to cover both hemispheres, and generates a normal view as well as orthogonal and stereographic views.
dualfisheyevideo (DFV)	<p>Reads a video clip which contains two full-circular fisheye images (assumed as equidistance projection) to cover both hemispheres, and generates a normal view as well as orthogonal and stereographic views.</p>  <p>(*13)</p>
verticalfisheye (VF)	Reads a full-circular fisheye image (assumed as equidistance projection) covering the upper hemisphere (*13), and dynamically generates a normal view as well as orthogonal and stereographic views of the hemisphere.

## Ryubin's Flash Panorama Player, User's Guide

	 <p>(*13)</p>
verticalfisheyevideo (VfV)	Video version of the verticalfisheye playmode.
fish2rect (F2R)	<p>Reads a full-circular fisheye image (assumed as equidistance projection) (*10) , and dynamically generates an equirectangular view of the image (*14).</p>  <p>(*14)</p>
fish2rectvideo (F2RV)	Video version of the fish2rect playmode.
verticalfish2rect (VF2R)	<p>Reads a full-circular fisheye image (assumed as equidistance projection) covering the upper hemisphere (*13), and dynamically generates an equirectangular view of the image (*15).</p>  <p>(*15)</p>
verticalfish2rectvideo (VF2RV)	Video version of the verticalfish2rect playmode.
cylinder (CY)	<p>Reads a cylindrical still image (a still image created using cylindrical projection) (*16a), and dynamically generates a normal (rectilinear) view of the still image.</p>



	 <p>(*16a)</p>
cylindervideo (CYV)	<p>Reads a cylindrical video image (a video image created using cylindrical projection) (*16b), and dynamically generates a normal (rectilinear) view of the video image.</p>  <p>(*16b)</p>
doughnut (D)	<p>Reads a doughnut-shaped still image (i.e., annular, ring or omni type of still image usually taken by using a bowl-shaped mirror lens) (*17), and dynamically generates a normal (rectilinear) view, as well as orthogonal and stereographic views of the still image.</p>
doughnutvideo (DV)	<p>Reads a doughnut-shaped video image (i.e., annular, ring or omni type of video image usually taken by using a bowl-shaped mirror lens) (*17), and dynamically generates a normal (rectilinear) view, as well as orthogonal and stereographic views of the video image. Video version of the doughnut playmode.</p>  <p>(*17)</p>
doughnut2rectvideo(D2RV)	<p>Reads a doughnut-shaped video image (i.e., annular, ring or omni type of video image usually captured by using a bowl-shaped mirror lens) (*17), and dynamically generates an equirectangular view of the image.(*15)</p>

## User Interface

The user interface of RFP player, incorporating interfaces for controlling panorama images and video clips, takes full advantage of the capabilities of Adobe Flash platform including the ideas of button object and the context menu, as well as such common user interfaces as mouse and keyboard used in every general personal computing environment.

(Table/Figure/Chart 1)

Panorama control				
OPERATION	Mouse	Keyboard	Context menu (valid only when the item is activated)	Controller (Built-in, or XML-based or SWF based external controllers)
To pan (increase / decrease yaw)	horizontal drag	right/left arrow key		right/left button
To tilt (increase / decrease pitch)	vertical drag	up/down arrow key		up/down button
To zoom in (increase FOV)	forward roll of wheel	"A" key Shift key		plus button or equivalent
To zoom out (decrease FOV)	backward roll of wheel	"Z" key Ctrl key		minus button or equivalent
To pan 90 degrees (increase / decrease yaw by 90 degrees)				double-right/double-left button or equivalent
To go back to home position (resets yaw, pitch, fov to initial/default values)		Home key	" <a href="#">Home Position</a> " item (or "reset viewing experience" item)	home button or equivalent
To show / hide viewing information (system info such as yaw, pitch, fov values)			"View info" item	system information button or equivalent (toggle)
To show / hide user- defined information				user information button or equivalent (toggle)
To enter / exit full screen mode		Insert key to enter, Esc key to exit	"Full screen" item	full-screen button or equivalent (toggle)
To show / hide hotspot(s)			"Hotspots" item	hotspot button or equivalent

## Ryubin's Flash Panorama Player, User's Guide

To start / stop auto panning			"Start Auto Panning" item	start-auto button to start, stop-auto button to stop
To enable / disable smooth filter			"Smoothing Filter" item	smooth-filter button or equivalent (toggle)
To show / hide user-definable panels (or images)			"User definable panels" item	
To change projection mode			"Sphere context" item	projection-mode button or equivalent
To change fisheye viewing mode			"Fisheye context" item	
<b>Video/Sound Control</b>				
<b>OPERATION</b>	<b>Mouse</b>	<b>Keyboard</b>	<b>Context menu (valid only when the item is activated)</b>	<b>Controller (built-in, SWF based or XML-based external ones)</b>
To pause video/sound				pause button or equivalent
To play (resume) video/sound				play button or equivalent
To rewind video/sound				rewind button or equivalent
To change video/sound playing position				position bar/ loaded bar or equivalent
To enable (turn on) sound				turn-on button or equivalent
To disable (turn off) sound				turn off button or equivalent

## Simple approach using built-in controller

### Basic setup

If you just want to play with panoramic images or motion pictures without any extra capability over the default features of RFP player, we recommend you to take the approach of using the built-in controller.

Let's take the "sphere" playmode as an example for explaining the simple approach.

If you [click here](#) to download a file named "sample.zip" and extract the zip file, you will obtain the following files and folders:

- sample.html (file)
- js (folder)
- pano\_image (folder)
- sound (folder)
- swf (folder)
- system\_images (folder)

If you run the sample.html using your favorite browser, you can view the following stage window with the built-in controller at the bottom:



The sample.html includes the following line of code:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>RyubinPanoPlayer</title>
<script type="text/javascript">AC_FL_RunContent = 0;</script>
<script src="js/AC_RunActiveContent.js" type="text/javascript"></script>
</head>
<body bgcolor="#333333" >
<script type="text/javascript">
AC_FL_RunContent(
'codebase', 'http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=10,0,0,0',
'pluginspage', 'http://www.macromedia.com/go/getflashplayer',
'bgcolor', '#000000',
'width', '640',
'height', '480',
'id', 'RyubinPanorama',
'movie', 'swf/RyubinPanoPlayer5',
'FlashVars', 'playmode=sphere&internal_ctrl=yes&img_path=pano_image/sphere.jpg&sound_path=sound/my_sound01.mp3
&cursor_path=system_images/my_cursor.png&fov=120&yaw=0&pitch=0&smooth=yes&seg_lock=medium',
'allowFullScreen', 'true',
'allowScriptAccess', 'always'
); //end AC code
</script>
</body>
</html>
```

By changing the parameters in this html file, you can customize the basic setup to meet your needs. For example, if you would like to change the width and/or the height of the stage, replace **640** and/or **480** with your preferred number(s).

If you would like to play your own panorama image whose name is “my\_panorama.jpg”, place the image file in the pano\_image folder and then change the string of “**sphere.jpg**” to “my\_panorama.jpg”.

If you do not need the sound, just eliminate the string of “**sound\_path=sound/my\_sound01.mp3&**”.

For details about the meanings of the “FlashVars” parameters, refer to the section titled “[FlashVars \(HTML\) Parameters](#)”.

As the concept of the basic setup is common to all of the playmodes, you could easily examine your own images or video clips with the RFP player.

If you would like to publish and play the panorama online, upload all the files and folders you created or customized based on the basic setup onto your Web server without changing the hierarchy of the folders (directories).

## Sophisticated approach using XML files

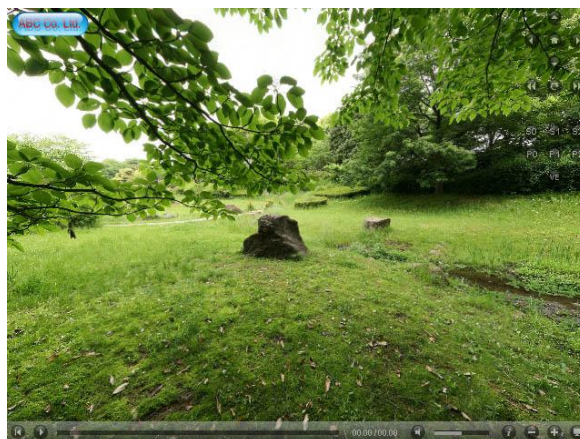
### Basic setup

If you would like to play with panoramic images or motion pictures with full capabilities such as the customizable controller over the default features of the RFP player, we recommend you to take the approach of using the XML files.

Let's take the "sphere" playmode as an example again for explaining this approach. If you [click here](#) to download a file named "sample.zip" and extract the zip file, you will obtain the following files and folders:

- sample.html (file)
- js (folder)
- object\_images (folder)
- pano\_image (folder)
- sound (folder)
- swf (folder)
- system\_images (folder)
- xml (folder)

If you run the sample.html using your favorite browser, you can view the following stage window with the customized controllers at the bottom and the top-right corner:





## Ryubin's Flash Panorama Player, User's Guide

The sample.html includes the following line of code:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>RyubinPanoPlayer</title>
<script type="text/javascript">AC_FL_RunContent = 0;</script>
<script src="js/AC_RunActiveContent.js" type="text/javascript"></script>
</head>
<body bgcolor="#333333" >
<script type="text/javascript">
AC_FL_RunContent(
'codebase', 'http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=10,0,0,0',
'pluginspage', 'http://www.macromedia.com/go/getflashplayer',
'bgcolor', '#333333',
'width', '640',
'height', '480',
'id', 'RyubinPanorama',
'movie', 'swf/RyubinPanoPlayer5',
'FlashVars', 'playmode=sphere&internal_ctrl=no&xml_path=xml/sphere_CC.xml',
'allowFullScreen', 'true',
'allowScriptAccess', 'always'
); //end AC code
</script>
</body>
</html>
```

Please find this HTML calls “[sphere\\_CC.xml](#)” by using the parameter of “[xml\\_path=](#)”.

And the [sphere\\_CC.xml](#) includes the following line of code:

```
<?xml version = '1.0'?>
<my_params>
<stage>
<sound_controller xml_path="xml/IntegratedCtrl_640.xml" posx="" posy="" alignx="center" popup="yes"
out_alpha="0.5" />
<panorama_controller xml_path="xml/ArrowButtons.xml" posx="" posy="" aligny="top" init_show="yes" popup="yes"
out_alpha="0.5" />
<cursor path="system_images/my_cursor.png" />
</stage>
<context_menu>
<panorama_controller show_item="yes" on_caption="Show Arrow Buttons" off_caption="Hide Arrow Buttons"/>
<sound_controller show_item="yes" />
<full_screen show_item="yes" on_caption="Enter fullscreen" off_caption="Exit fullscreen" />
<u_context1 show_item="yes" on_caption="Show Message" off_caption="Hide Message" />
<about_me my_name=" Put your brand or company name here" my_link="http://www.ryubinpanorama.com" />
</context_menu>
<user_panel>
<u_context1 path="system_images/my_message.jpg" posx="0" posy="0"/>
<logo path="system_images/my_logo.png" posx="" posy="" anchor="tl"/>
</user_panel>
<play_objects>
<image path="pano_image/sphere.jpg" />
<info text="Canon EOS Kiss Digital X + Canon Fisheye 15mm + PTGui" size="16" align="center" bold="no" italic="yes"
height="30" posy="-40" alignx="center" />
<sound path="sound/my_sound02.mp3" init_loudness="0.5" init_start="no" />
```

```
<view init_filter="yes" init_fov="90" lock_nadirs="yes" limit_vertical="no" top_limit="54" bottom_limit="57"  
seg_lock="medium" quality_lock="no" unlock_for_non_rectilinear="yes" />  
<auto init_start="no" />  
</play_objects>  
</my_params>
```

A set of XML tags, elements and attributes coded in the [sphere\\_CC.xml](#) defines how the RFP player in the “sphere” playmode operates and behaves.

In this way, by using the XML file, you can make the most of the advanced capabilities of the RFP player.

For details about the XML specifications, consult with the section titled “[Specifications of the primary XML file](#)”.

As for typical usages of major tags, elements and attributes, read the section titled “[Typical usages of major XML tags, elements and attributes](#)”



## Typical usages of major XML tags, elements and attributes

### Specifying panorama image, video, sound to play

The most significant element in the XML file is the <image> element in the <play\_object> tag, which is used to specify the path and the image file or video clip to play by using the RFP player.

In the following example, “image/my\_panorama.jpg” is specified as the panorama image to be played.

```
<?xml version = '1.0'?>
<my_params>
.
.
.
<play_objects>
  <image path="image/my_panorama.jpg" />
</play_objects>
</my_params>
```

In the following example, “video/my\_panorama.flv” is specified as the panorama video to be played.

```
<?xml version = '1.0'?>
<my_params>
.
.
.
<play_objects>
  <image path="video/my_panorama.flv" />
</play_objects>
</my_params>
```

In the following example, “sound/my\_sound.mp3” is specified as the sound to be played together with the panorama image: “my\_panorama.pg”.

```
<?xml version = '1.0'?>
<my_params>
  <play_objects>
    <image path="image/my_panorama.jpg" />
    <sound path="sound/my_sound.mp3" />
  </play_objects>
</my_params>
```

Please note that the sound can only be played with a still image, and can not coexist with the sound track of a video clip.

For details, consult with the section titled "[Specifications of the primary XML file](#)".

## User definabe mouse cursor

By using <cursor> element with the "path" attribute within the <stage> tag, you can replace the system cursor (mouse pointer) with the image created by yourself.

```
<?xml version = '1.0'?>
<my_params>
  <stage>
    <cursor path="sys_img/my_cursor.png" />
  </stage>
  .
  .
  .
</my_params>
```

In the able example, the "path" attribute refers the image named "my\_cursor.png" located in the "sys\_img" directory, as an alternative image for the system cursor. For details, consult with the section titled "[Specifications of the primary XML file](#)".

## Initial viewing experience

There might be a case where you want to change or configure the viewing experience when the viewer is initiated. Each of the RFP player allows you to specify the initial value or condition when it starts, by using several attributes implemented for the <view /> element under the <play\_objects> tag.

The following list summarizes a typical set of values/conditions of the viewing experience that can be defined by the user.

Initial viewing experience	Attribute (for <view /> under <play_objects> tag) to change the default setting	Default setting
----------------------------	---	-----------------

yaw	init_yaw [ <i>number in degree</i> ]	0
pitch	init_pitch [ <i>number in degree</i> ]	0
fov	init_fov [ <i>number in degree</i> ]	60 for still playmodes 100 for video playmodes
Smoothing filter	init_filer [yes/no]	no

```
<?xml version = '1.0'?>
.
.
<play_objects>
  <view init_filter="yes" init_fov="90" init_yaw="90" init_pitch="10" />
</play_objects>
</my_params>
```

For details, consult with the section titled “[Specifications of the primary XML file](#)”.

## Panorama controller

By adding panorama controller, which is an external xml file delivered by us or developed by yourself, to the panorama stage; you will have an additional way, on top of the mouse and keyboard interfaces, to control your viewing experience, i.e., fov (field of view), pitch, yaw and others.

```
<?xml version = '1.0'?>
<my_params>
  <stage>
    <panorama_controller xml_path="xml/ArrowButtons.xml" posx="10" posy="10" alignx="right" aligny="top"
    init_show="yes" popup="yes" out_alpha="0.5" init_show="yes" />
  </stage>
  <context_menu>
    <panorama_controller show_item="yes" on_caption="Show Panorama Controller" off_caption="Hide Panorama
    Controller"/>
  </context_menu>
.
</my_params>
```

You will find an element named `<panorama_controller>` within the `<stage>` tag. This is the way how to add the panorama controller using XML. In this example, the XML refers the panorama controller whose name is ArrowButtons.xml which is located in the “xml” directory.

In addition, if you add an item to the context menu by using the `<panorama_controller>` element as shown in the above example, you can show or hide the panorama controller via the context menu while the player is running.

For details, consult with the section titled "[Specifications of the primary XML file](#)".

## Sound controller (still only)

You can play a sound while a still panorama image is being displayed on the stage, by using the `<sound />` element and the "path" attribute within the `<play_objects>` tag, as already described in the section of "[Specifying panorama image, video, sound to play](#)". While the sound file referenced by the "path" attribute is successfully loaded, the sound will automatically start without the need of the sound controller, by adding the sound controller to the stage, you can control the sound in such way like pausing, resuming, rewinding, adjusting the loudness.

You can add the sound controller by using the `<sound_controller />` element together with the "path" attribute within the `<stage>` tag, as shown in the following example.

```
<?xml version = '1.0'?>
<my_params>
  <stage>
    <sound_controller xml_path="xml/IntegratedCtrl_640.xml" posx="" posy="" alignx="center" popup="yes"
    out_alpha="0.5" />
  </stage>
  <context_menu>
    <sound_controller show_item="yes" on_caption="Show Sound Controller" off_caption="Hide Sound Controller"/>
  </context_menu>

  <play_objects>
    <sound path="sound/my_sound.mp3" />
  </play_objects>
</my_params>
```

In the above example, a sound file named "my\_sound.mp3" under the directory of "sound" will be played, and the sound controller which is an external XML file named "IntegratedCtrl\_640.xml" will be added to the stage.

In addition, if you add an item to the context menu by using the `<sound_controller>` element in the `<context_menu>` tag as shown in the above example, you can show or hide the sound controller via the context menu while the player is running.

For details, consult with the section titled "[Specifications of the primary XML file](#)"

## Video controller (video only)

You can play a video clip by specifying the designated video file via the `<sound />` element and the "path" attribute within the `<play_objects>` tag, as already described in the section of "[Specifying panorama image, video, sound to play](#)". While the video file referenced by the "path" attribute is successfully loaded, the video can automatically start without the need of the video, by adding the video controller to the stage, you can control the video clip in such way like pausing, resuming, rewinding, adjusting the position to play and the loudness of the sound if exists.

You can add the video controller by using the `<video_controller />` element together with the "path" attribute within the `<stage>` tag, as shown the following example.

```
<?xml version = '1.0'?>
<my_params>
  <stage>
    <video_controller xml_path="xml/IntegratedCtrl.xml" posx="10" posy="10" alignx="center" aligny="bottom"
    popup="yes" out_alpha="0.5" />
  </stage>
  <context_menu>
    <video_controller show_item="yes" on_caption="Show Video Controller" off_caption="Hide Video Controller"/>
  </context_menu>

  <play_objects>
    <image path="video/my_panorama.flv" />
  </play_objects>
</my_params>
```

In the above example, a video file named "my\_panorama.flv" under the directory of "video" will be played, and the video controller which is an external XML file named "IntegratedCtrl.xml" will be added to the stage.

In addition, if you add an item to the context menu by using the `<video_controller>` element in the `<context_menu>` tag as shown in the above example, you can show or hide the video controller via the context menu while the player is running.

For details, consult with the section titled "[Specifications of the primary XML file](#)"

## User definable panels (or images)

By using the <user\_panel> tag and <u\_conext1> and <u\_context2> elements along with the “path” attribute, you can add a user definable panel or an image graphics or even a Flash movie to the panowindow of the RFP player.

```
<?xml version = '1.0'?>
<my_params>
.
.
  <user_panel>
    <u_context1 path="sys_img/instruction01.gif" posx="0" posy="-10"/>
    <u_context2 path="sys_img/message01.jpg" posx="0" posy="0"/>
  </user_panel>
  <context_menu>
    <u_context1 show_item="yes" on_caption="Show Instruction" off_caption="Hide Instruction" />
    <u_context2 show_item="yes" on_caption="Show Message" off_caption="Hide Message" />
  </context_menu>
.
.
</my_params>
```

In this example, “instraction01.jpg” and “message01.jpg” in the “sys\_img” directory are specified. These panels can be separately made visible or invisible while the RFP player is running by adding respective item within the context menu in the way as shown above.

For details, consult with the section titled “[Specifications of the primary XML file](#)”.

## Defining panowindow

The RFP player allows you to define a sub-pane where the projected panorama image or video clip is actually presented. The sub-pain which we call “panowindow” is created inside the stage area that is specified by the “width” operand and the “height” operand of “AC\_FL\_RunContent” JavaScript function and/or <object> tag and/or <embed> tag used in the HTML file.

The following table shows how the stage size is defined using the HTML codes.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>RyubinPanoPlayer</title>
<script type="text/javascript">AC_FL_RunContent = 0;</script>
<script src="js/AC_RunActiveContent.js" type="text/javascript"></script>
```

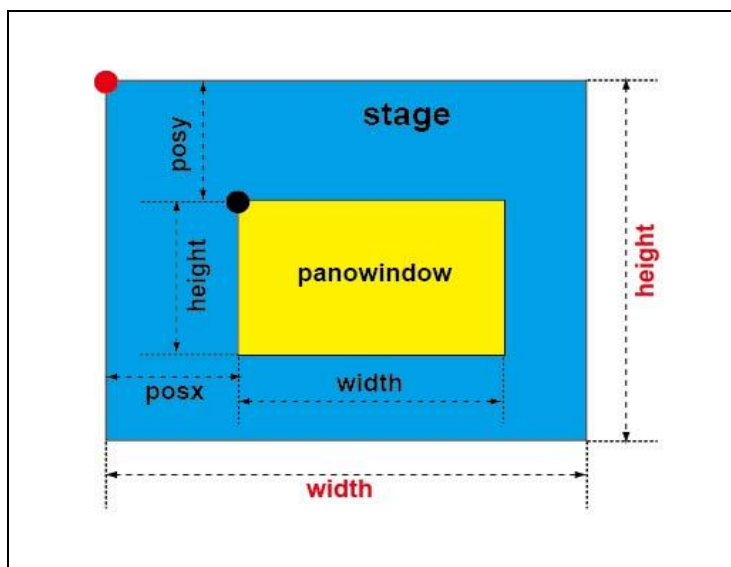
```
</head>
<body bgcolor="#333333" >
<script type="text/javascript">
AC_FL_RunContent(
'codebase', 'http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=10,0,0,0',
'pluginspage', 'http://www.macromedia.com/go/getflashplayer',
'bgcolor', '#333333',
'width', '640',
'height', '480',
'id', 'RyubinPanorama',
'movie', 'swf/RyubinPanoPlayer5',
'FlashVars', 'playmode=sphere&internal_ctrl=no&xml_path=xml/sphere_CC.xml',
'allowFullScreen', 'true',
'allowScriptAccess', 'always'
); //end AC code
</script>
</body>
</html>
```

To define a “panowindow” within the stage, use the `<panowindow />` element of `<stage>` tag, along with “fullstage”, “width”, “height”, “posx”, “posy” attributes. Please set `fullstage="no"` before defining a “panowindow” inside the stage. The size of the “panowindow” is set by “width” and “height” attributes and the relative position of the “panowindow” to the stage is set by “posx” and “posy” attributes.

The following XML sample shows how the “panowindow” is defined using the `<panowindow />` element and the associated attributes.

```
<my_params>
  <stage>
    <panowindow fullstage="no" width="540" height="360" posx="50" posy="60" />
  </stage>
  .
  .
</my_params>
```

The relative position of the panowindow is defined by the horizontal and the vertical distance of the panowindow to the stage, as illustrated below. The “width” and the “height”, both in red colored characters and are defined by the HTML codes as previously described, represent the size of the stage. The “width” and the “height”, both in black colored characters and are defined by the XML tags/elements/attributes as previously described, represent the size of the panowindow. And the “posx” and “posy”, both in black colored characters and are defined by the XML tags/elements/attributes as previously described, represent the relative position of the panowindow to the stage. For details, consult with the section titled “[Specifications of the primary XML file](#)”.



## User definable skin

RFP player allows you to use a user definable skin which is a background image that can be placed on the stage area. Please use `<user_panel>` tag, `<skin />` element along with associated attributes such as “path”, “posx”, and “posy”, to place a skin to the stage as shown in the example below, in which `<stage>` tag, `<panowindow />` element, and the associated attributes are also used to define a sub-pane “panowindow” where the RFP player presents projected panorama images. This is a typical approach of placing a panowindow over the skin image defined by the `<skin />` element within the `<user_panel>` tag.

For details, consult with the section titled “[Specifications of the primary XML file](#)”.

```
<?xml version = '1.0'?>
<my_params>
  <stage>
    <panowindow fullstage="no" width="540" height="360" posx="50" posy="60" />
  </stage>
  .
  .
  <user_panel>
    <skin path="sys_img/my_skin.png" />
  </user_panel>
  .
  .
</my_params>
```



## User definable logo image

RFP player allows you to show a user definable small image or a logo image representing your company or workshop over the panowindow, a sub-pane area created inside the stage to present the actual panorama image or movie clip. Please use `<user_panel>` tag, `<logo />` element along with associated attributes such as “path”, “posx”, and “posy”, to place a logo image to the panowindow as shown in the example below.

For details, consult with the section titled “[Specifications of the primary XML file](#)”.

```
<?xml version = '1.0'?>
<my_params>
.
  <user_panel>
    <logo path="sys_img/my_logo.png" anchor="tl" />
  </user_panel>
.
.
</my_params>
```

## User definable system messages

The RPF player allows you to use your own character strings as alternatives to the default character strings for several system messages, by using respective attributes such as “fov\_max”, “loading”, and others within the `<caption />` element, as shown below.

For details, consult with the section titled “[Specifications of the primary XML file](#)”.

```
<?xml version = '1.0'?>
<my_params>
  <stage>
    <caption fov_max=" Stop zooming out" fov_min=" Stop zooming in" loading="now loading panorama" />
  </stage>
.
.
</my_params>
```

## Customizing the auto panning behaviors

### Managing auto panning operation

The users of RFP player are able to manage the auto panning operation by using a set of attributes; “init\_start“, “pan\_speed“, “tilt\_amplitude“, and “tilt\_cycle” in the <auto /> element of the <play\_objects> tag. Use the “init\_start” attribute to define whether or not the auto panning is to be started immediately after the player is loaded. The default setting is init\_start=”no”, meaning that the auto panning will not start when the player is loaded. Use the “pan\_speed” attribute to change the panning speed. For “tilt\_cycle” and “tilt\_amplitude” attributes, please see the next section.

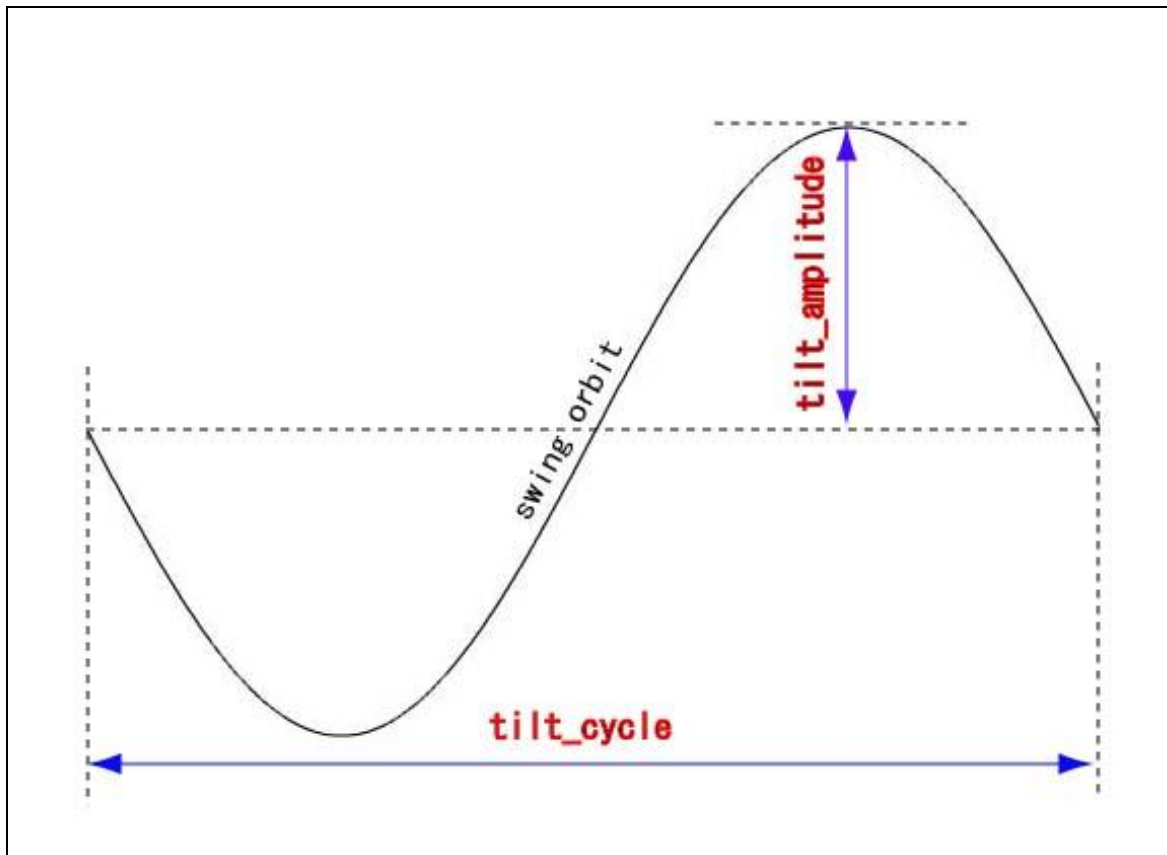
For details, consult with the section titled “[Specifications of the primary XML file](#)”.

```
<?xml version="1.0" ?>
<my_params>
.
.
  <play_objects>
    <auto init_start="yes" pan_speed="0.05" tilt_amplitude="45" tilt_cycle="150" />
  </play_objects>
.
.
</my_params>
```

### Enabling sinusoidal panning

This is a unique feature to the RFP player, allowing the user to automatically pan the image with a sinusoidal swing having a combination of cycle and amplitude that can be specified by the user. Please use “tilt\_cycle” and “tilt\_amplitude” attributes in the <auto /> element of the <play\_objects> tag to specify the values of cycle and amplitude in degree. The speed of panning can be specified by “pan\_speed” attribute.

For details, consult with the section titled “[Specifications of the primary XML file](#)”.



## Customizing the image projection

### Cropping the image

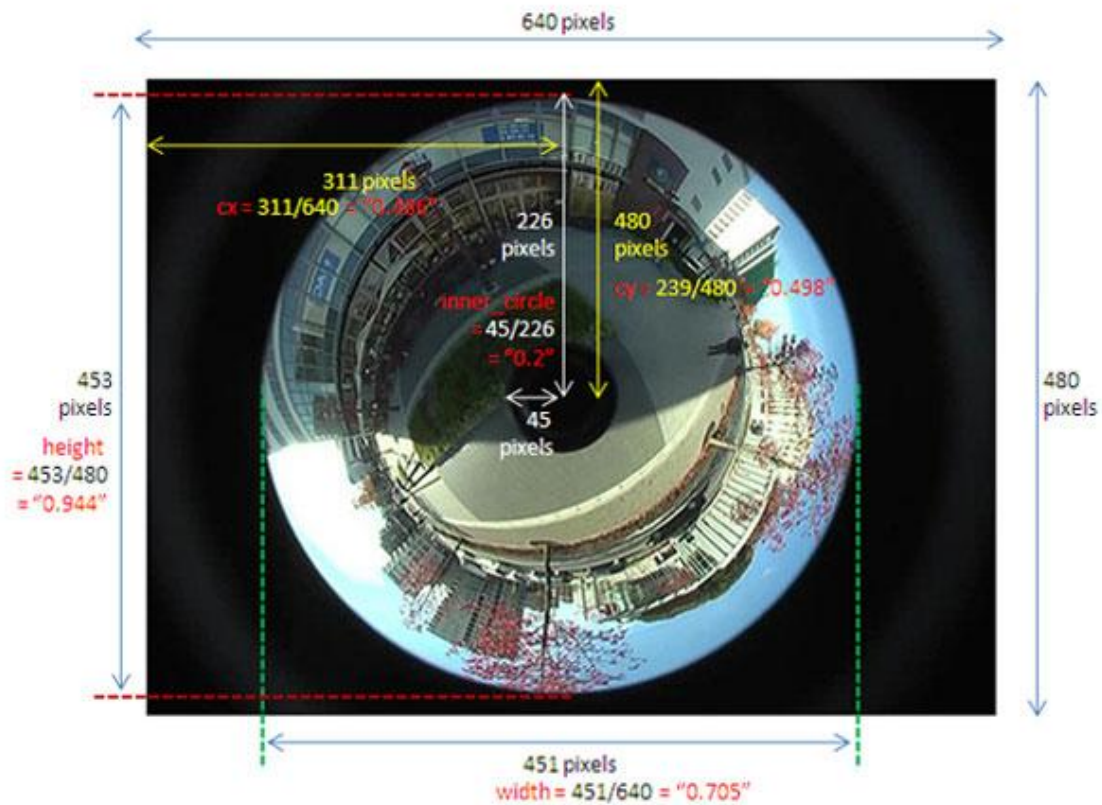
Among all, the RFP player in a plymode such playmodes like fisheye, fisheyevideo, doughnut, or doughnutvideo reads an image data that includes a circular shaped shot taken though a full-fisheye lens, a panoramic annular lens, or a single shot mirror, as shown below.



Each of those playmodes internally crops the necessary part of the image to recreate an intended panoramic view, i.e., a rectilinear, orthogonal or stereographic projection of the image. In order to tell what part of the image to crop, the RFP player leverages the XML scheme and implements `<crop />` element and other associated attributes. For details, consult with the section titled “[Specifications of the primary XML file](#)”.

```
<?xml version = '1.0'?>
<my_params>
.
.
  <play_objects>
    <image path="video/my_doughnut_video_01.flv" />
    <crop width="0.705" height="0.944" cx="0.486" cy="0.498" inner_circle="0.2"/>
    <projection elevation="46" depression="62"/>
  </play_objects>
.
.
</my_params>
```

### <crop> example



### Projecting the image

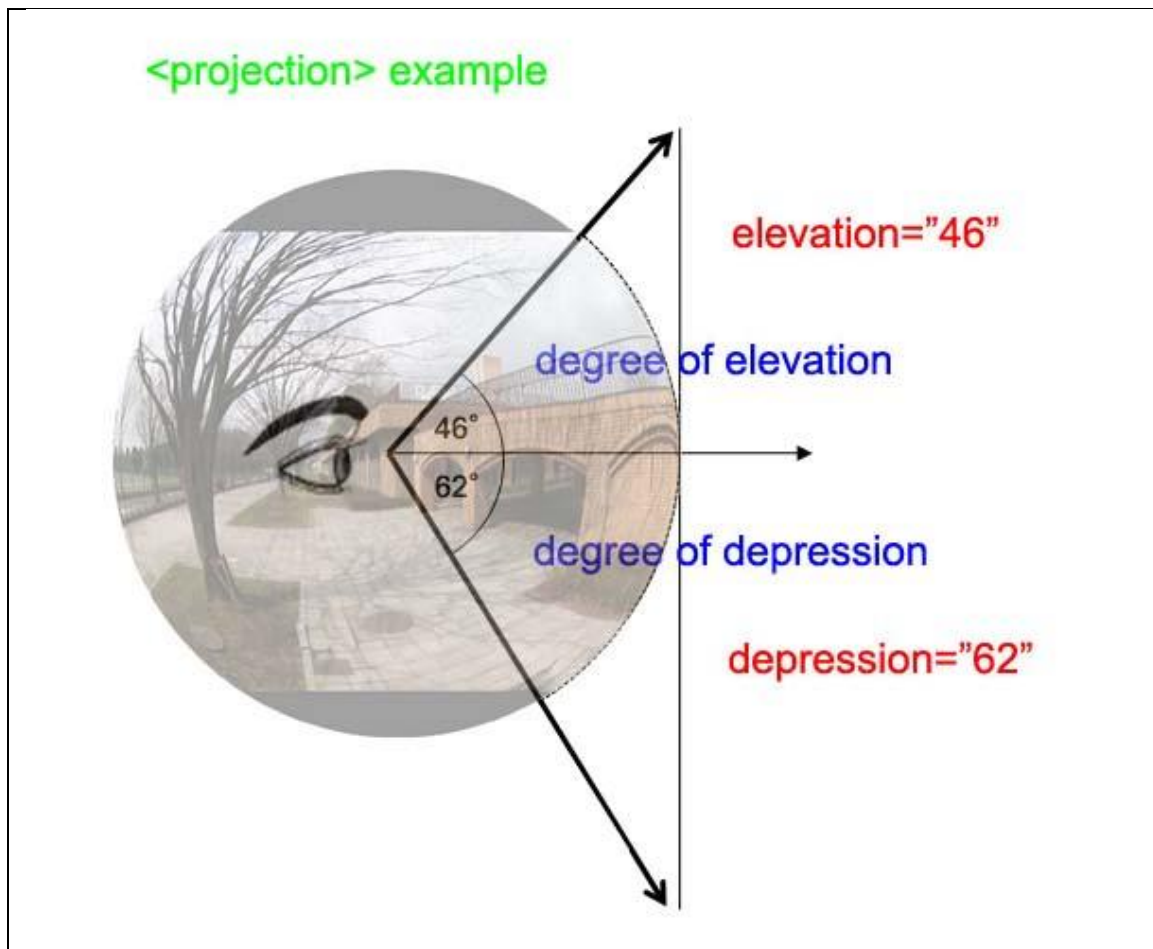
RFP player can be categorized by their type of image and the shape of textured object, as follows:

Category	Playmode	Type of image (image projection)	Textured object or 3D model	Typical vertical travel
Cube	cube2x3 cube3x2 cube6 cube24 cube2x3video cube3x2video	Cubic	Cube	Full 180 degree
Sphere	sphere sphere8	Full-equirectangular	Sphere	Full 180 degree

	spherevideo	Partial-equirectangular (Landscape)		Partial
Fisheye	fisheye dualFisheye fisheyevideo	Full fisheye	Sphere	Full 180 degree
	verticalfisheye verticalfisheyevideo	Full fisheye		Full 90 degree
Doughnut	doughnut doughnutvideo	Doughnut		Partial
Cylinder	cylinder cylinderVideo	Cylindrical	Cylinder	Partial
Flat	flat flatVideo simpleflat simpleflatvideo	Full-equirectangular or partial-equirectangular or whatever type of image	Flat plane (or 2D plane)	(Not applicable) (hv_ratio is an alternative)
	fish2rect fish2rectvideo	Full fisheye		
	verticalfish2rect verticalfish2rectvideo	Full fisheye		
	doughnut2rectvideo	Doughnut		

Among all of the image types assumed for the RFP player, “Doughnut”, “Cylindrical”, and “Partial-rectangular” are the typical image types whose vertical travels are partial or limited, meaning that the assumed span of the vertical travel should be explicitly specified. In other words, when using those types of image, upper limit of the travel or elevation and the lower limit of the travel or depression have to be explicitly specified so that the designated image can be appropriately wrapped onto the texturing object; “sphere” or “cylinder”.

The following figure shows how an image is wrapped onto the surface of a sphere as a texture using the specified values of elevation and depression.



The degrees of elevation and depression can be set by using `<projection />` element within the `<play_objects>` tag, along with the associated attributes as follows. For details, consult with the section titled "[Specifications of the primary XML file](#)".

```
<?xml version = '1.0'?>
<my_params>
.
.
.
  <play_objects>
    <image path="video/my_doughnut_video_01.flv" />
    <crop width="0.705" height="0.944" cx="0.486" cy="0.498" inner_circle="0.2"/>
    <projection elevation="46" depression="62"/>
  </play_objects>
.
.
</my_params>
```

## Handling partial-equirectangular image

The sphere playmode and the spherevideo playmode are originally intended to accept full-equirectangular image, of which the vertical travel fully spans across 180 degree. Some users wanted to handle partial-equirectangular image that does not cover full 180 degree vertical-wise. There are two ways to recognize the value of the intended vertical travel; implicit approach and explicit approach.

As default, the sphere playmode and the spherevideo playmode implicitly handle the partial-equirectangular image by using the ratio of vertical aspect and horizontal aspect (i.e. height/width of the image) of the image to calculate the value of the vertical travel as follows:

If vertical-horizontal ratio is larger than or equal to 0.5, the vertical travel is set to 180 degree; else the vertical travel is set to the multiple of vertical-horizontal ratio and 360 degree. Taking an image whose size is 1000 pixels in width and 250 pixels in height for example, the vertical travel is set to  $(250/1000) \times 360 = 90$  degree. In the default case, the value of vertical travel is evenly split into the value of elevation and depression. In the above example, both the depression and the depression become equally 45 degree.

However, some other uses wanted to handle a partial-equirectangular image whose elevation and depression are not even. To meet this requirement, we have implemented a new attribute named "implicit" in the <projection /> element, allowing users to explicitly, directly and separately specify the values of elevation and depression by using the attributes of "elevation" and "depression".

For details, consult with the section titled "[Specifications of the primary XML file](#)".

A usage example follows:

```
<?xml version = '1.0'?>
<my_params>
.
.
  <play_objects>
    <image path="image/my_panorama.jpg" />
    <projection implicit="no" elevation="52" depression ="75"/>
  </play_objects>
.
.
</my_params>
```



## Specifying "hv\_ratio" for Flat (2D) playmodes

For the RFP player categorized as "Flat" that use a flat or 2D plane as texturing object, we did not implement the concept of 3D projection, meaning that the value of elevation and depression are not applied to those playmodes. Instead, the "Flat" playmodes including flat, flatvideo, fish22rect, fish2rectvideo, veritcalfish2rect, verticalfish2rectvideo and doughnut2rect, an internal parameter named "hv\_ratio" is used to shape the viewing aspect ratio, meaning that the width-height ratio of the actual image is neglected but the default or explicitly specified "hv\_ratio" is used to determine the ratio of horizontal aspect and the vertical aspect, i.e. the viewing aspect.

playmode	Default value of "hv_ratio"
flat	2.0
flatvideo	2.0
fish2rect	1.0
fish2rectvideo	1.0
verticalfish2rect	4.0
verticalfish2rectvideo	4.0
doughnut2rectvideo	3.5

A usage example follows:

```
<?xml version = '1.0'?>
<my_params>
  <play_objects>
    <image path="pano_image/flat.jpg" />
    <view hv_ratio="2.5" />
  </play_objects>
</my_params>
```

For details, consult with the section titled "[Specifications of the primary XML file](#)".

## Limiting vertical viewing-direction

There are several attributes usable in the <view /> element of the <play\_objects> tag to limit or restrict the vertical viewing-directions or the range of vertical travel.

Please use "lock\_nadirs" to lock the vertical viewing-direction at the top and bottom nadirs.

Please use "lock\_vertical" to lock the vertical viewing-direction to the initial pitch value (i.e. pitch value will not be changed). Default is "no" (not locked).

Please use "limit\_vertical" to limit the vertical viewing-direction within a span. Please note that the limit\_vertical is forced to "no" when lock\_nadirs="yes" or lock\_vertical="yes" is specified. Default is "no" (not limited).

Please use "top\_limit" to specify the top limit for the span of vertical viewing-direction in degree. Default is +90. This attribute works only when limit\_vertical="yes" is specified.

Please use "bottom\_limit" to specify the bottom limit for the span of vertical viewing-direction in degree. Default is +90. This attribute works only when limit\_vertical="yes" is specified.

For details, consult with the section titled "[Specifications of the primary XML file](#)".

### Limiting horizontal viewing-direction

There are several attributes usable in the <view /> element of the <play\_objects> tag to limit or restrict the horizontal viewing-directions.

Please use "limit\_horizontal" to limit the vertical viewing-direction within a span. Default is "no" (not limited).

Please use "right\_limit" to specify the right limit for the span of right viewing-direction in degree. Default is +90. This attribute works only when limit\_horizontal="yes" is specified.

Please use "left\_limit" to specify the left limit for the span of horizontal viewing-direction in degree. Default is +90. This attribute works only when limit\_horizontal="yes" is specified.

For details, consult with the section titled "[Specifications of the primary XML file](#)".

### Flipping the image

Please use "flip\_vertical" and/or "flip\_horizontal" in the <view /> element of the <play\_objects> tag to show the image in vertically and/or horizontally flipped manner.

For details, consult with the section titled "[Specifications of the primary XML file](#)".

## Hotspots

The RFP player allows you to comile multiple panorama images, videos, and even external Web pages into a set of collection by using a technique called “hotspot”. A hotspot is usually placed or overridden on an object such as a door, window, or building within a panorama image. When clicking on a hotspot, you will be invited to other panorama image, video, or external Web page associated to the hotspot.

In order to implement this mechanism , the RFP player implements `<hot_spot>` tag, `<add />` element, `<behavior />` as well as associated attributes such as “yaw”, “pitch”, “img\_path”, “caption”, “action”, “dest”, and “target” for the `<add />` element, and “init\_show” attribute for `<behavior />` element.

The “yaw” and “pitch” attributes are used to define the position of the hotspot to be placed. The “img\_path” is used to refer an image data to be used as a hotspot. The “caption” is used to define a character string made visible when the mouse cursor goes over the hotspot.

The “action” attribute defines the action to be taken when a hotspot is clicked on. Three types of action are implemented; “roam”, “warp”, and “link. Both roam and link call other XML file which defines a set of viewing experiences to be displayed when the hotspot is clicked on. If “roam” is specified, the RFP player reads only the `<play_objects>` tag and its associated elements and attributes to form the next viewing experiences. If “warp” is specified, the player reads full XML content including the `<play_objects>` tags so that all of the settings including the exsting hotspot definition can be replaced by the new one. In both cases; “roam” and “warp”, the designated XML file can be specified by the “dest” attribute.

If “link” is specified, the player reads an external or internal Web page directed by the “dest” attribute. The “target” attribute is associated to the “link” attribute, and is used to define an HTML window where the designated Web page is displayed.

For details, consult with the section titled “[Specifications of the primary XML file](#)”.

## Customizing the context menu

### Adding an item to reset viewing experience (home position)

You may occasionally want to go back to the original viewing direction or the field of view while you are playing a panorama image or video. As already shown in the section of [“User Interface”](#), this operation can be performed by means of keyboard, controller and the context menu item. Here you will see the way to do it by using a context menu item. As shown below, if you add the `<home_pos />` element with “show\_item” attribute you can add an item to reset viewing experience such as yaw, pitch and the field of view. Only when the attribute “show\_item is set to “yes”, the item becomes visible. For details, consult with the section titled [“Specifications of the primary XML file”](#).

```
<?xml version = '1.0'?>
<my_params>
.
.
  <context_menu>
    <home_pos show_item="yes" caption="Reset viewing experience" />
  </context_menu>
.
.
</my_params>
```

### Adding an item to enable/disable smoothing filter

Place the `<smoothing />` element within the `<context_menu>` tag to enable/disable the smoothing filter via the context menu. The item becomes visible only when you specify `show_item="yes"` within the `<smoothing />` element.

For details, consult with the section titled [“Specifications of the primary XML file”](#).

```
<?xml version = '1.0'?>
<my_params>
.
.
  <context_menu>
    <smoothing show_item="yes" on_caption="Enable smoothing" off_caption="Disable smoothing" />
  </context_menu>
.
.
</my_params>
```

## Adding an item to enter/exit full-screen mode

Place the `<full_screen />` element within the `<context_menu>` tag to enter/exit full-screen mode via the context menu. The item becomes visible only when you specify `show_item="yes"` within the `<full_screen />` element.

For details, consult with the section titled "[Specifications of the primary XML file](#)".

```
<?xml version = '1.0'?>
<my_params>
.
.
  <context_menu>
    <full_screen show_item="yes" on_caption="Enter full-screen" off_caption="Exit full_screen" />
  </context_menu>
.
.
</my_params>
```

## Adding an item to start/stop auto panning

Place the `<start_auto />` element within the `<context_menu>` tag to start/stop the auto panning operation via the context menu. The item becomes visible only when you specify `show_item="yes"` within the `<start_auto />` element.

For details, consult with the section titled "[Specifications of the primary XML file](#)".

```
<?xml version = '1.0'?>
<my_params>
.
.
  <context_menu>
    <start_auto show_item="yes" on_caption="Start auto pan" off_caption="Stop auto pan" />
  </context_menu>
.
.
</my_params>
```

## Adding an item for "sphere context"

By adding `<sphere_context>` tag along with the associated elements and attributes in the XML file, you will be able to change the type of projection among "normal (rectilinear)", "globe (orthogonal)" and "bipolar (stereographic)". These context menu items work only for the playmode such as sphere, spherevideo, sphere8, fisheye, fisheyevideo, verticalfisheye, verticalfisheyevideo, doughnut, and doughnutvideo.

Each of those elements such as <normal>, <globe> and <bipolar> corresponds to the projection mode of “normal (rectilinear)”, “globe (orthogonal)” and “bipolar (stereographic)” respectively.

For details, consult with the section titled “[Specifications of the primary XML file](#)”.

An example follows:

```
<?xml version = '1.0'?>
<my_params>
.
.
  <sphere_context>
    <normal show_item="yes" on_caption="--> Rectilinear projection" off_caption=" Rectilinear projection" />
    <globe show_item="yes" on_caption="--> Orthogonal projection" off_caption=" Orthogonal projection" />
    <bipolar show_item="yes" on_caption="--> Stereographic projection" off_caption=" Stereographic projection" />
  </sphere_context>
.
.
</my_params>
```

### Adding an item for “fisheye context”

This capability is applicable to the playmode of fisheye and fisheyevideo. The RFP player in the fisheye or fisheyevideo playmode reads a full-circular fisheye image (assumed as equidistance projection), and dynamically generates a normal view as well as orthogonal and stereographic views. While a full circular fisheye image usually covers only a half of the space (hemisphere), RFP player in the fisheye playmode is capable of internally duplicating the image serial-wise or mirror-wise and dynamically generates a pseudo full-space view.

By placing <fisheye context> tag in the XML file, you will be able to change the way of using the designated full-fisheye or hemisphere image as a texture when wrapping it onto the 3D sphere model in the player. Three elements such as <single>, <double> and <mirror> can be used within the <fisheye\_context> tag to specify one of three way of texture usage. In the “single” mode, the texture is used to cover a hemisphere of the 3D sphere model. In the “double” mode, the texture is duplicated and concatenated serial wise and used to cover full-sphere of the 3D sphere model. In the “mirror” mode, the texture is duplicated and concatenated mirror-wise and used to cover full-sphere of the 3D sphere model.

For details, consult with the section titled “[Specifications of the primary XML file](#)”.

```
<?xml version = '1.0'?>
<my_params>
.
.
  <fisheye_context>
    <single show_item="yes" on_caption="" off_caption="" />
    <double show_item="yes" on_caption="" off_caption="" />
    <mirror show_item="yes" on_caption="" off_caption="" />
  </fisheye_context>
.
.
</my_params>
```

### Adding an item to show/hide system information (viewing experience)

By placing `<show_info />` element within the `<context_menu>` tag as shown below, you can add an item to show/hide a set of values of viewing experience such as yaw, pitch and the field of view on the top-left corner of the panowindow. Only when you specify `show_item="yes"` within the `<show_info />` element, the context menu item becomes visible, and its function is enabled.

For details, consult with the section titled [“Specifications of the primary XML file”](#).

```
<?xml version = '1.0'?>
<my_params>
.
.
  <context_menu>
    <show_info show_item="yes" on_caption="Show viewing info" off_caption="Hide viewing info" />
  </context_menu>
.
.
</my_params>
```

### Customizing the loading indicator (still playmode only)

If you add `<user_loader>` tag and the associated elements and attributes to the XML, you can customize the loading indicator by using three graphics images you create yourself respectively for the three components; blinking caption, progress bar, and background.

```
<?xml version = '1.0'?>
<my_params>
.
.
.
  <user_loader>
    <caption path="sys_img/my>LoadingCap.png" />
    <indicator path="sys_img/my>LoadingInd.png" />
    <background path="sys_img/my>LoadingBack.png" />
  </user_loader>
.
.
.
</my_params>
```

In the above example, the `<caption />` element and the “path” attribute specify the location and the name of the image data to be used for the caption, which continues blinking while the designated panorama image is being loaded. And, the `<indicator />` element and the “path” attribute specify the location and the name of the image data to be used as the progress bar, the length or width of the image will show the relative progress status of the image being loaded. The `<background />` element and the “path” attribute is used to refer the location and the name of the image data to be used as the background of the whole loading indicator.

For details, consult with the section titled “[Specifications of the primary XML file](#)”.

### Balancing the quality and performance (video only)

As already stated in the section of “[Pros & Cons](#)”, the Adobe Flash platform implements an abstraction layer to absorb the differences among operating systems; Windows, Mac and Linux, in order to make itself a pervasive platform available to unlimited Internet audiences. This nature in turn gives you a range of challenges when handling bits and bytes of the images.

In fact, especially when working with panoramic motion pictures, you might face a situation where you need to compromise or make a balance among such visual factors as screen size, stage size, video size (width x height), FPS (frame per second), encoding bitrates and others until you obtain a practical or compromised condition for your working environment or designated client platform.



On top of the visual factors listed above, there is a significant factor which affects the quality and performance of panorama experience. The RFP player internally segment the intended image into a number of small triangular meshes so that so-called “Affine Transformation” can be applied to each of those meshes, instead of applying the coordinate transformation directly to all of the pixels of the image. In a sense, the approach of taking the “Affine” as a coordinate transformation technique is a process of compromise or balancing quality and performance. The smaller the size of the mesh, in other words, the larger the number of the segments, the higher the quality or the lower the performance is. In this term, the RFP player implement a unique approach where the number of the segment is set lower when the panorama experience such as yaw, pitch or FOV is being changed, so that the image movement becomes faster and lighter. We call this approach “segment release”.

In addition, the RFP player implement another unique approach of defining two levels of rendering quality; “high” and “low”. Similarly to the segment setup, the level of the rendering quality is set lower when the panorama experience is being changed to make the movement faster, while it is set higher during the stay-on status. We call this approach “(rendering) quality release”.

However, users' conditions inevitably vary on their environment, needs, and preferences and we found it needed to allow users to manage those conditions themselves by implementing some additional attributes to help explicitly define the those setups.

The following table shows how we implemented two additional attributes “seg\_lock” and “quality\_lock” in the <view /> element under <play\_objects> to help users manage the above two factors “segment” and “rendering quality” by themselves.

For details, consult with the section titled “[Specifications of the primary XML file](#)”.

seg_lock   v	quality_lock		
	“high”	“low”	“no” or unspecified (default) Rendering quality is released
“high”	Highest quality Lowest performance		Segment is not released Rendering quality is released
“medium”			
“low”		Lowest quality Highest performance	
“no” or unspecified (default)	Segment is released Rendering quality is not released		Default setup Both segment and rendering quality are

Segment is released		released
---------------------	--	----------

## Configuring video starting and preloading behavior (video only)

Currently the RPF viewers implement the approach of “progressive download” when remotely playing panoramic video clips. In short, the “progressive download” is a unilateral method to download motion pictures from an ordinary HTTP server to the user with an option to play the video simultaneously while the video is being downloaded. A well-known challenge with these simultaneous play-download approach whether it is “progressive download” or pure “bilateral streaming”, is that you have to consider situations where the bandwidth is not sufficient to meet the data transfer requirement to freely play the intended video sequence without losing the original playing speed. There might be several approaches to compromise these challenges, such as to dynamically reduce the frame rate to prioritize the bandwidth requirement, or to pause the video until sufficient amount of video is buffered so that you can keep the original frame rate unchanged, or to eventually wait until whole video footage is downloaded to escape from the influence of the bandwidth.

In order to allow users to manage these situations, the RFP player implement an XML element named `<behavior>` under `<play_objetcs>` XML tag as well as several parameters configurable via XML attributes. We categorized these parameters in two;

- A) Flags to setup the video starting behavior
- B) Values to manage video preloading behavior

For setting up the starting behavior of a video clip, there are four flags represented by respective attributes to be used within the `<behavior />` element:

1) **load\_on\_start**= "yes/no" (default = “yes”) ---> abbreviated as LOS

This flag defines if the video-load operation is to be started immediately at the first frame of the video clip.

2) **wait\_for\_preload** = "yes/no" (default = "no")---> abbreviated as WFP

This flag defines if the video-play operation is to be suspended until the minimum pre-load condition is met.

3) **pause\_on\_start**="yes/no" (default = "no") ---> abbreviated as POS

This flag defines if the video-play operation is to be started immediately at the first frame of the video clip.

4) **start\_button**="yes/no" (default = "no") ---> abbreviated as SB

This flag defines if the start button is to be made visible and activated.

**Usage example (represents the Case 6 in the following table) :**

```
<play_objects>
<image path="wait_for_selection" video_repeat="no" />
<behavior load_on_start="no" wait_for_preload="yes" pause_on_start="no" start_button="no"
preload_minimum="0.2" preload_start="0.95" preload_stop="0.8" />
<view init_fov="120" init_yaw="" init_pitch="" init_filter="no" seg_lock="medium" lock_nadirs="yes" />
</play_objects>
```

**Combination of the above four flags and the intended starting behavior.**

LOS	WFP	POS	SB	starting behavior	Case #
"yes"	"yes"	"yes" or "no" (neglected)	"yes" or "no" (neglected)	1) Video-load will automatically start but video-play pauses at the first frame. 2) The video-play will automatically start when the minimum pre-load condition is met. 3) POS setting will be neglected. 4) The start button and the PressStartButton() JS (JavaScript) function are disabled. Even if SB="yes" is set, the start button will not be visible.	Case 1
	"no"	"yes"	"yes"	1) Video-load will automatically start but video-play pauses at the first frame. 2) The video-play will immediately start when the start button is pressed or the PressStartButton() JS (JavaScript) function is called.	Case 2
			"no"	1) Video-load will automatically start but video-play pauses at the first frame. 2) The video-play will immediately	Case 3

				start only when the PressStartButton() JS (JavaScript) function is called.	
		"no"	"yes" or "no" (neglected)	1) Both video-load and video-play will automatically start w/o any intervention. 2) The start button and the PressStartButton() JS (JavaScript) function are disabled. Even if SB="yes" is set, the start button will not be visible.	Case 4 (default behavior)
"no"	"yes"	"yes" or "no" (neglected)	"yes"	1) Both video-load and video-play pause at the first frame. 2) The video-load will immediately start when the start button is pressed or the PressStartButton() JS (JavaScript) function is called. 3) The video-play will automatically start when the minimum pre-load condition is met. 4) POS setting will be neglected.	Case 5
			"no"	1) Both video-load and video-play pause at the first frame. 2) The video-load will immediately start only when the PressStartButton() JS (JavaScript) function is called. 3) The video-play will automatically start when the minimum pre-load condition is met 4) POS setting will be neglected.	Case 6
	"no"	"yes" or "no" (neglected)	"yes"	1) Both video-load and video-play pause at the first frame. 2) The video-load as well as video-play will immediately start when the start button is pressed or the PressStartButton() JS (JavaScript) function is called. 3) POS setting will be neglected.	Case 7
			"no"	1) Both video-load and video-play pause at the first frame. 2) The video-load as well as video-play will immediately start only when the the PressStartButton() JS (JavaScript) function is called. 3) POS setting will be neglected.	Case 8

For setting up the values to manage video preloading behavior, there are three parameters represented by respective attributes to be used within the <behavior /> element:

1) **preload\_minimum**= "*number*" (default value = "0.1")

This value defines the minimum preload condition. When wait\_for\_preload="yes", the video\_play operation will pause until the minimum preload condition is met. The value represents the ratio of the preloaded video length per total video length.

2) **preload\_start** = "*number*" (default value = "0.95")

This value defines the condition to start "video-preload" status, i.e. "preload\_start" condition where "video\_play" operation pauses to wait until "preload-stop" condition is met. The value represents the ratio of the already played video length (or video position, play head position) per the preloaded video length.

3) **preload\_stop**="number" (default value = "0.8")

This value defines the condition to top "video-preload" status, i.e. "preload\_stop" condition where "video\_play" operation resumes. The value represents the ratio of the already played video length (or video position, play head position) per the preloaded video length.

**Usage example (represents the Case 6 in the following table) :**

```
<play_objects>
<image path="wait_for_selection" video_repeat="no" />
<behavior load_on_start="no" wait_for_preload="yes" pause_on_start="no" start_button="no"
preload_minimum="0.2" preload_start="0.95" preload_stop="0.8" />
<view init_fov="120" init_yaw="" init_pitch="" init_filter="no" seg_lock="medium" lock_nadirs="yes" />
</play_objects>
```

For details, consult with the section titled "[Specifications of the primary XML file](#)".

## Allowing cross domain access

There might be a case where you would like to use a media file which is located at a different domain from the domain where your viewer resides. In that case, you need to set cross\_domain attribute to "yes". The cross\_domain attribute is implemented as one of the attributes to <image /> element and <sound /> element within <play\_objects> tag.. For details, consult with the section titled "[Specifications of the primary XML file](#)".

Even when you get approved from your viewer to access other domain by specifying `cross_domain="yes"`, the actual access will be denied unless the target domain allows it via its security policy. For more information about the concept of the cross-domain security policy, please refer

[http://www.adobe.com/devnet/articles/crossdomain\\_policy\\_file\\_spec.html](http://www.adobe.com/devnet/articles/crossdomain_policy_file_spec.html)

## Allowing the use of JavaScript API

When using JavaScript API, two flags should be taken care of. First, in the HTML file, you have to set `allowScriptAccess` to “always” or “sameDomain” to allow a JavaScript code to access the viewer as shown the following example. If you set “sameDomain”, the script access is allowed only when the HTML file and the viewer (SWF) are located at the same domain, otherwise the script access is always allowed.

Sample HTML follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<META http-equiv="Content-Style-Type" content="text/css">
<TITLE>PanoFisheyeFlashVideo_v20</TITLE>
<SCRIPT type="text/javascript">AC_FL_RunContent = 0;</SCRIPT><SCRIPT src="js/AC_RunActiveContent.js"
type="text/javascript"></SCRIPT> </HEAD>
<BODY bgcolor="#333333" >
<SCRIPT type="text/javascript">
AC_FL_RunContent(
'codebase', 'http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=10,0,0,0',
'pluginspage', 'http://www.macromedia.com/go/getflashplayer',
'bgcolor', '#333333',
'width', '400',
'height', '300',
'id', 'RyubinPanorama',
'movie', 'RyubinPanoPlayer5',
'FlashVars', 'xml_path=xml/doughnutvideo_CC.xml',
'allowFullScreen', 'true',
'allowScriptAccess', 'always'
); //end AC code
</SCRIPT>
</BODY>
</HTML>
```

The second point you need to take care is the attribute of `allow_script_access` which is implemented as an attribute for `<security />` element within the `<stage>` tag. This attribute should be set “yes” before the JavaScript API becomes workable.

Sample XML follows:

```
<?xml version="1.0" ?>
<my_params>
.
.
  <stage>
    <security allow_script_access="yes" />
  </stage>
.
.
</my_params>
```

For details, consult with the section titled “[Specifications of the primary XML file](#)”.

## JavaScript API

The RFP player implements several application program interfaces accessible from a JavaScript code.

Please note that the attribute of “allow\_script\_access” should be set “yes” to make these JavaScript APIs operable as described in the section of “[Allowing the use of JavaScript API](#)”.

Here is the complete list of JavaScript API currently usable with the RFP player.

JavaScriptAPI	Parameter/ Return value	Function
<b>set_viewdirection</b> (TgtYaw, TgtPitch)	<b>TgtYaw:</b> target yaw value in degree <b>TgtPitch:</b> target pitch value in degree	Sets or changes the view direction using the specified yaw and pitch values.
<b>set_videopos</b> (TgtPos)	<b>TgtPos:</b> target video position in second	Sets or changes the current video playing position (i.e. head position) using the specified position value.
<b>press_start_button</b> ()	none	Emulates the press operation of the start button.
<b>play_video</b> ()	none	Plays the current video clip.
<b>pause_video</b> ()	none	Pauses the current video clip.
<b>rewind_video</b> ()	none	Rewinds the current video clip.
<b>get_status</b> (): status[4]	<b>status[0]:</b> video length in second <b>status[1]:</b> current	Returns the length of the video being played, current video position as well as

	video position in second <b>status[2]</b> :current yaw value in degree <b>status[3]</b> :current pitch value in degree	current yaw and pitch values.
<b>enable_video_repeat()</b>	none	Enables video-repeat. The current video clip repeatedly played without stooping at the end of the footage.
<b>disable_video_repeat()</b>	none	Disables the video-repeat operation. The current video clip stops playing at the end of the footage.
<b>loadImg</b> (image)	<b>image</b> :URL referring an image file to be loaded and played	Loads the image specified by "image" operand. The specified image starts playing if it's found and successfully loaded.
<b>loadVideo</b> (video)	<b>video</b> :URL referring a video file to be loaded and played	Loads the video specified by "video" operand. The current video clip suspends and the specified video starts playing if it's found and successfully loaded.
<b>loadXML</b> (xml)	<b>xml</b> : URL referring an XML file to be loaded	Loads the XML specified by the "xml" operand. If it's found and successfully loaded, and if the play objects specified in the XML is found and successfully loaded, the currently playing object will stop running and the specified play object will start running.
<b>ConveyStatus</b> (length, play_pos, yaw, pitch)	<b>length</b> : video length in second <b>play_pos</b> : current video position (i.e. head position) in second <b>yaw</b> : current yaw value in degree <b>pitch</b> : current pitch value in degree	This is a passive function periodically called by the SWF (i.e. the video viewer) if appropriately defined in the JavaScript code.



A sample usage of JavaScript API follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<META http-equiv="Content-Style-Type" content="text/css">
<TITLE>playmode=spherevideo</TITLE>
<SCRIPT type="text/javascript">AC_FL_RunContent = 0;</SCRIPT><SCRIPT src="js/AC_RunActiveContent.js"
type="text/javascript"></SCRIPT>
</HEAD>
<BODY bgcolor="#888888" >
<SCRIPT type="text/javascript">
AC_FL_RunContent(
'codebase', 'http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=10,0,0,0',
'pluginspage', 'http://www.macromedia.com/go/getflashplayer',
'bgcolor', '#000000',
'width', '400',
'height', '200',
'id', 'RyubinPanorama',
'name', 'PanoramaPlayer',
'movie', 'RyubinPanoPlayer5',
'FlashVars', 'playmode=spherevideo&internal_ctrl=yes&xml_path=xml/PanoSphereFlashVideo_07.xml',
'allowFullScreen', 'true',
'allowScriptAccess', 'always'
); //end AC code
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript1.1">
<!--
var flashPanoramaPlayer;

function initialize() {
    /* Check if the browser is IE. If so, flashPanoramaPlayer is window.PanoramaPlayer.
    Otherwise, it's window.document.PanoramaPlayer. The PanoramaPlayer is the id assigned to <object> and <embed>
    tags. */
    var ie = navigator.appName.indexOf("Microsoft") != -1;
    flashPanoramaPlayer = (ie) ? window['PanoramaPlayer'] : document['PanoramaPlayer'];
}

function SetViewDirection(TgtYaw,TgtPitch) {
    initialize();
    flashPanoramaPlayer.set_viewdirection(TgtYaw, TgtPitch);
}

function SetVideoPos(TgtPos) {
    initialize();
    flashPanoramaPlayer.set_videopos(TgtPos);
}

function PressStartButton() {
    initialize();
    flashPanoramaPlayer.press_start_button();
}

function PlayVideo() {
    initialize();
    flashPanoramaPlayer.play_video();
}

function PauseVideo() {
    initialize();
```

```

        flashPanoramaPlayer.pause_video();
    }

    function RewindVideo() {
        initialize();
        flashPanoramaPlayer.rewind_video();
    }
    function GetStatus() {
        initialize();
        var my_status = flashPanoramaPlayer.get_status();
        document.interfaceform.length2.value = my_status[0];
        document.interfaceform.curr_pos2.value = my_status[1];
        document.interfaceform.curr_yaw2.value = my_status[2];
        document.interfaceform.curr_pitch2.value = my_status[3];
    }

    function ConveyStatus(length, play_pos, yaw, pitch) {
        document.interfaceform.length.value = length;
        document.interfaceform.curr_pos.value = play_pos;
        document.interfaceform.curr_yaw.value = yaw;
        document.interfaceform.curr_pitch.value = pitch;
    }
    function SelectVideo(video) {
        initialize();
        flashPanoramaPlayer.loadVideo(video);
    }
    function EnableVideoRepeat() {
        initialize();
        flashPanoramaPlayer.enable_video_repeat();
    }
    function DisableVideoRepeat() {
        initialize();
        flashPanoramaPlayer.disable_video_repeat();
    }
}
//--->
</SCRIPT>
<br>Video clips: Courtesy of <a href="http://www.globalvision.ch/" target="_blank">GlobalVision Communication
</a>
<form name="interfaceform">
<hr>
<B>ConveyStatus() (passive function)</B><br>
Leng: <input type="text" name="length" size="6" value="0" /> Pos: <input type="text" name="curr_pos" size="6"
value="0" />
Yaw: <input type="text" name="curr_yaw" size="3" value="0" /> Pitch: <input type="text" name="curr_pitch" size="3"
value="0" />
<hr>
<input type="button" value="GetStatus()" onClick="GetStatus()"/><br>
Leng: <input type="text" name="length2" size="6" value="0" /> Pos: <input type="text" name="curr_pos2" size="6"
value="0" />
Yaw: <input type="text" name="curr_yaw2" size="3" value="0" /> Pitch: <input type="text" name="curr_pitch2" size="3"
value="0" />
<hr>
Pos: <input type="text" name="tgt_pos" size="6" value="0" /> <input type="button" value="SetVideoPos()"
onClick="SetVideoPos(document.interfaceform.tgt_pos.value)"/>
<hr>
Yaw: <input type="text" name="tgt_yaw" size="3" value="0" /> Pitch: <input type="text" name="tgt_pitch" size="3"
value="0" /> <input type="button" value="SetViewDirection()"
onClick="SetViewDirection(document.interfaceform.tgt_yaw.value,document.interfaceform.tgt_pitch.value)"/>
<hr>
<input type="button" value="PressStartButton()" onClick="PressStartButton()"/>
<hr>
<input type="button" value="PlayVideo()" onClick="PlayVideo()"/> <input type="button" value="PauseVideo()"
onClick="PauseVideo()"/> <input type="button" value="RewindVideo()" onClick="RewindVideo()"/>

```

```
<hr>
<input type="button" value="EnableVideoRepeat()" onClick="EnableVideoRepeat()"/> <input type="button"
value="DisableVideoRepeat()" onClick="DisableVideoRepeat()"/>
<hr>
<input type="button" value="SelectVideo(A)" onClick="SelectVideo('video/Cannes2008A.flv')"/>
<input type="button" value="SelectVideo(B)" onClick="SelectVideo('video/Cannes2008B.flv')"/><br><br>
<input type="button" value="SelectVideo(C)" onClick="SelectVideo('video/Cannes2008C.flv')"/>
<input type="button" value="SelectVideo(D)" onClick="SelectVideo('video/Cannes2008D_mod02.flv')"/>
<hr>
To test FNTF function: <input type="button" value="SelectVideo(X)" onClick="SelectVideo('video/Cannes2008X.flv')"/>
<hr>
</form>
</BODY>
</HTML>
```

## Publishing .flv files on the Web server (video)

In order for .flv files to be correctly handled on your server, the MIME type for .flv file-type should be appropriately configured.

For a Linux-based server, make sure the following line is included in the .htaccess file.  
AddType video/x-flv .flv

When serving .flv files off of a Windows Server 2003 (or any other Windows server) , follow the steps below to add .flv MIME type in IIS

- 1) Select the site to configure in IIS, right click and select "Properties"
- 2) Under HTTP Headers Tab, select "File Types" under the MIME Map section and select "New Type"
- 3) Type ".flv" as the associated extension and "video/x-flv" as the content type.
- 4) Select "OK" to complete.

For details, consult with your Web site or hosting administrator.

## Specifications of FlashVars (HTML) parameters

PARAMETER And value	FUNCTION	APPLICABLE PLAYMODE(S)
<b>playmode</b> =[ <i>playmode</i> ]	Specifies the play mode with which the player operates.	N/A
<b>Internal_ctrl</b> =[yes no]	Specifies if the internal or built-in controller is used. If yes is specified, the player runs with the pre-defined internal or built-in controller. Default is no.	all playmodes
<b>xml_path</b> =[ <i>path/file_name</i> ]	Specifies path and name or URL of the primary XML file, which includes a range of predefined XML tags, elements and attributes to define an operational behavior of the player. The specification of the primary XML files is described in the following section titled " <a href="#">XML tags for primary XML file</a> ".	all playmodes
<b>img_path</b> =[ <i>path/file_name</i> ]	This parameter is equivalent to the <b>path</b> attribute in the <image /> element in the <play_objects> XML tag.	Refer XML specifications
<b>video_path</b> =[ <i>path/file_name</i> ]	This parameter is equivalent to the <b>img_path</b> parameter when used in video playmodes.	Refer XML specifications
<b>cursor_path</b> =[ <i>path/file_name</i> ]	This parameter is equivalent to the <b>path</b> attribute in the <cursor /> element in the <stage> XML tag	Refer XML specifications
<b>fov</b> =[ <i>number</i> ]	This parameter is equivalent to the <b>init_fov</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>yaw</b> =[ <i>number</i> ]	This parameter is equivalent to the <b>init_yaw</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>pitch</b> =[ <i>number</i> ]	This parameter is equivalent to the <b>init_pitch</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>view_mode</b> =[ <i>number</i> ]	This parameter is equivalent to the <b>init_view_mode</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>hv_ratio</b> =[ <i>number</i> ]	This parameter is equivalent to the <b>hv_ratio</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>smooth</b> =[yes no]	This parameter is equivalent to the <b>init_filter</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>lock_nadirs</b> =[yes no]	This parameter is equivalent to the <b>lock_nadirs</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>lock_vertical</b> =[yes no]	This parameter is equivalent to the <b>lock_vertical</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>lock_fov</b> =[yes no]	This parameter is equivalent to the <b>lock_fov</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications

<b>limit_vertical</b> =[yes no]	This parameter is equivalent to the <b>limit_vertical</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>top_limit</b> =[number]	This parameter is equivalent to the <b>top_limit</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>bottom_limit</b> =[number]	This parameter is equivalent to the <b>bottom_limit</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>limit_horizontal</b> =[yes no]	This parameter is equivalent to the <b>limit_horizontal</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>right_limit</b> =[number]	This parameter is equivalent to the <b>right_limit</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>left_limit</b> =[number]	This parameter is equivalent to the <b>left_limit</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>seg_lock</b> =[high medium low no]	This parameter is equivalent to the <b>seg_lock</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>quality_lock</b> =[high low no]	This parameter is equivalent to the <b>quality_lock</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications
<b>elevation</b> =[number]	This parameter is equivalent to the <b>elevation</b> attribute in the <projection /> element in the <play_objects> XML tag.	Refer XML specifications
<b>depression</b> =[number]	This parameter is equivalent to the <b>depression</b> attribute in the <projection /> element in the <play_objects> XML tag.	Refer XML specifications
<b>proj_implicit</b> =[yes no]	This parameter is equivalent to the <b>implicit</b> attribute in the <projection /> element in the <play_objects> XML tag.	Refer XML specifications
<b>proj_type</b> =[number]	This parameter is equivalent to the <b>proj_type</b> attribute in the <projection /> element in the <play_objects> XML tag.	Refer XML specifications
<b>disable_wheel</b> =[yes no]	This parameter is equivalent to the <b>disable_wheel</b> attribute in the <view /> element in the <play_objects> XML tag.	Refer XML specifications

## Specifications of the primary XML file

### <stage> tag

#### <panowindow /> element

ATTRIBUTE And value	FUNCTION	APPLICABLE PLAYMODE(S)
<b>fullstage</b> =["yes" "no"]	Specifies if the full area of the stage is used as a "pano-window" where panoramic images or videos are drawn. The size of the stage is defined by HTML statements such as "width" and "height" elements in the <object /> HTML tag. When using a JavaScript code instead such as AC_RunActiveContent.js, "width" and "height" parameters are used to define the stage size. When "no" is specified, subsequent attributes such as <b>posx</b> , <b>posy</b> , <b>width</b> and <b>height</b> are used to define the "pano-window" where to draw the panoramic images or videos. Default is "yes". This attribute is useful when using a user defined skin that can be specified by the <skin /> element in the <user_panel> tag.	all playmodes
<b>posx</b> =["number(0)"] <b>posy</b> =["number(0)"]	Valid only when <b>fullstage</b> ="no". The player uses the values set by <b>posx</b> and <b>posy</b> to determine where to locate the top-left corner of the pano-window in the stage area. The value of <b>posx</b> should be set in pixel representing the horizontal distance between of the left edges of the stage and the pano-window, and the value of <b>posy</b> should be set in pixel representing the vertical distance between of the top edges of the stage and the pano-window. The default value for both attributes is "0".	all playmodes
<b>width</b> =["number(400)"] <b>height</b> =["number(300)"]	Valid only when <b>fullstage</b> ="no". The player uses the pixel values set by <b>width</b> and <b>height</b> to define the size of the pano-window. Default values for <b>width</b> and <b>height</b> are "400" and "300" respectively.	all playmodes

#### <cursor /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>path</b> =["path/file_name"]	Specifies path and name or URL for a user-definable cursor, which is an external image file to be used in place of the system cursor (mouse pointer) when the mouse is dragged in the panorama viewing area (i.e. pano-window) defined by <panowindow> tag.	all playmodes

## Ryubin's Flash Panorama Player, User's Guide

<b>movement_style</b> =["0" "1"]	When dragging on the image objects shown within the panowindow, some users might want to let them move opposite direction to the way the RyubinPanoPlayer originally does. In this case, specify <b>movement_style</b> ="1". Default setting is "0" for original movement style.	all playmodes
-------------------------------------	---	---------------

### <sandglass /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>path</b> =["path/file_name"]	Specifies path and name or URL for a user-definable sandglass, which is an external image file to be shown while the player is connecting to the net until the designated video file becomes ready for play.	video playmodes

### <panorama\_controller /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>xml_path</b> =["path/file_name"]	Specifies path and name or URL of an XML file, which includes a range of XML tags to define a graphical user interface of the player to control the viewing experiences such as fov, yaw, pitch and others. The XML file is called " <a href="#">XML-based customizable controller</a> ".	all playmodes
<b>path</b> =["path/file_name"]	Specifies path and name or URL for a panorama controller, which is an external SWF file created beforehand to control the viewing experiences such as yaw, pitch, fov and others. The SWF file is called "SWF-based external panorama controller". Please note that the <b>xml_path</b> and the <b>path</b> attributes are exclusive, meaning that the "XML-based customizable controller" and the "SWF-based external controller" cannot co-exist.	all playmodes
<b>posx</b> =["number(0)"] <b>posy</b> =["number(0)"]	Both attributes, to be set as pixel values, in combination specify the relative location to the original position of the panorama controller in the stage. Default values are zero. The original position of the controller is defined by the <b>alignx</b> and <b>aligny</b> attributes described below.	all playmodes
<b>alignx</b> =["top" "center" "bottom"] <b>aligny</b> =["top" "center" "bottom"]	Both attributes in combination specify the original location of the controller within the stage. Default settings are <b>alignx</b> ="right" and <b>aligny</b> ="bottom".	all playmodes
<b>init_show</b> =["yes" "no"]	Specifies if the panorama controller is enabled (i.e., displayed) when the player is launched. Default is "yes".	all playmodes
<b>popup</b> =["yes" "no"]	Specifies if the panorama controller appears in a pop-up way, meaning that the controller becomes visible (with 100% alpha value) when the mouse pointer moves over the controller, else the alpha will	all playmodes

	be down to the values specified by the <b>out_alpha</b> attribute described below. This attribute is applicable to the " <a href="#">XML-based customizable controller</a> " only. Default is "no".	
<b>out_alpha</b> =["number(0"] 0.0 <= value <= 1.0	Specifies the alpha value of the controller when the mouse pointer is out of the image of the controller. This attribute works only when the <b>popup</b> attribute is set "yes" as described above.	all playmodes
<b>show_after_load</b> =["yes" "no"]	Specifies if the "XML-based customizable controller" is shown after loading the designated panorama image file(s). Default is "no"; the controller is shown before loading the panorama image(s).	still playmodes

### <sound\_controller /> element

<sound\_controller /> element is intended to be used with a controller (XML based or SWF based) that supports sound-control capability such as play, pause and rewind as well as loudness or volume control of the sound. Therefore, this element is supposed to be used for play-modes that handle still images (i.e. still playmodes) along with a sound file (mp3 file).

Supported attributes are equivalent to those of the <panorama\_controller> element.

### <video\_controller /> element

<video\_controller /> element is intended to be used with a controller (XML based or SWF based) that supports video-control capability such as play, pause and rewind as well as loudness or volume control of the sound. Therefore, this element is supposed to be used for play-modes that handle video clips (i.e. video playmodes) that include a sound track).

### <captions /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>fov_max</b> =["Cstring"]	Specifies a character string to be used as an alternative to the default character string for showing the field of view reaches its maximum value. If you would like to eliminate the string, specify <b>fov_max</b> =" ". (use blank character(s)).	all playmodes
<b>fov_min</b> =["Cstring"]	Specifies a character string to be used as an alternative to the default character string for showing the field of view reaches its	all playmodes



	minimum value. If you would like to eliminate the string, specify <b>fov_min=""</b> . (use blank character(s)).	
<b>loading</b> =["Cstring"]	Specifies a character string to be used as an alternative to the default character string for showing the panorama image is being loaded.	still playmodes
<b>connecting</b> =["Cstring"]	Specifies a character string to be used as an alternative to the default character string for showing the net connection is underway.	all playmodes
<b>video_preloading</b> =["Cstring"]	Specifies a character string to be used as an alternative to the default character string for showing the video data is being preloaded.	video playmodes
<b>select_video_to_play</b> =["Cstring"]	Specifies a character string to be used as an alternative to the default character string for prompting to select a video clip to play.	video playmodes
<b>video_file_not_found</b> =["Cstring"]	Specifies a character string to be used as an alternative to the default character string for alerting the video clip cannot be found.	video playmodes
<b>image_file_not_found</b> =["Cstring"]	Specifies a character string to be used as an alternative to the default character string for alerting the image file cannot be found.	still playmodes
<b>roaming</b> =["Cstring"]	Specifies a character string to be used as an alternative to the default character string for showing the player is roaming to play another panorama image or video clip.	all playmodes
<b>warping</b> =["Cstring"]	Specifies a character string to be used as an alternative to the default character string for showing the player is warping to play another panorama image or video clip.	all playmodes
<b>hotspots_enabled</b> =["Cstring"]	Specifies a character string to be used as an alternative to the default character string for showing the hotspots are enabled and being displayed.	all playmodes

### <controller\_font /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>type_prog_ind</b> =["Cstring(Arial Narrow)"]	Specifies the type of font such as "Arial" and "Arial Narrow" to be used for the progress indicator defined in a "XML-based customizable controller". Default is "Arial Narrow"	all playmodes
<b>size_prog_ind</b> =["number(10)"]	Specifies the size of font to be used for the progress indicator defined in a "XML-based customizable controller". Default is "10".	all playmodes
<b>color_prog_ind</b> =["Cstring(FFFFFF)"]	Specifies the color of font to be used for the progress indicator defined in a "XML-based customizable controller". Default is "FFFFFF" (white). Specify "FF0000" for red, "00FF00" for green, and "0000FF" for blue.	all playmodes
<b>bg_prog_ind</b> =["yes" "no"]	Specifies whether or not to use a background for the font to be used for the progress indicator defined in a "XML-based customizable controller".	all playmodes

## Ryubin's Flash Panorama Player, User's Guide

	Default is "no".	
<b>bg_color_prog_ind</b> =["Cstring(FFFFFF)"]	Specifies the background color for the font to be used for the progress indicator defined in a "XML-based customizable controller". Default is "000000" (black). Specify "FF0000" for red, "00FF00" for green, and "0000FF" for blue.	all playmodes
<b>type_pos_ind</b> =["Cstring(Arial Narrow)"]	Specifies the type of font such as "Arial" and "Arial Narrow" to be used for the positional indicator defined in a "XML-based customizable controller". Default is "Arial Narrow"	all playmodes
<b>size_pos_ind</b> =["number(10)"]	Specifies the size of font to be used for the positional indicator defined in a "XML-based customizable controller". Default is "10".	all playmodes
<b>color_pos_ind</b> =["Cstring(FFFFFF)"]	Specifies the color of font to be used for the positional indicator defined in a "XML-based customizable controller". Default is "FFFFFF" (white). Specify "FF0000" for red, "00FF00" for green, and "0000FF" for blue.	all playmodes
<b>bg_pos_ind</b> =["yes" "no"]	Specifies whether or not to use a background for the font to be used for the positional indicator defined in a "XML-based customizable controller". Default is "no".	all playmodes
<b>bg_color_pos_ind</b> =["Cstring(FFFFFF)"]	Specifies the background color for the font to be used for the positional indicator defined in a "XML-based customizable controller". Default is "000000" (black). Specify "FF0000" for red, "00FF00" for green, and "0000FF" for blue.	all playmodes
<b>type_button</b> =["Cstring(Arial Narrow)"]	Specifies the type of font such as "Arial" and "Arial Narrow" to be used for the caption of a button defined in a "XML-based customizable controller". Default is "Arial Narrow"	all playmodes
<b>size_button</b> =["number(10)"]	Specifies the size of font to be used for the caption of a button defined in a "XML-based customizable controller". Default is "10".	all playmodes
<b>color_button</b> =["Cstring(FFFFFF)"]	Specifies the color of font to be used for the caption of a button defined in a "XML-based customizable controller". Default is "FFFFFF" (white). Specify "FF0000" for red, "00FF00" for green, and "0000FF" for blue.	all playmodes
<b>bg_button</b> =["yes" "no"]	Specifies whether or not to use a background for the font to be used for the caption of a button defined in a "XML-based customizable controller". Default is "no".	all playmodes
<b>bg_color_button</b> =["Cstring(FFFFFF)"]	Specifies the background color for the font to be used for the caption of a button defined in a "XML-based customizable controller". Default is "000000" (black). Specify "FF0000" for red, "00FF00" for green, and "0000FF" for blue.	all playmodes

### <security /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>allow_script_access</b> =["yes" "no"]	Specifies if the JavaScript API is made usable. Default is "no".	all playmodes

## <system\_msgs> tag

### <center\_msg /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>text</b> =["Cstring"]	Specifies the text string used as a system message displayed at the center of the pano-window. Default setting is "connecting.....".	all playmodes
<b>font</b> =["Cstring(Arial Narrow)"]	Specifies the type of font such as "Arial" and "Arial Narrow" to be used for the text of the center message, Default is "Arial"	
<b>bold</b> =["yes"] "no"]	Specifies if the bold font is used for the text of the center message. Default is "yes".	
<b>italic</b> =["yes"] "no"]	Specifies if the italic font is used for the text of the center message. Default is "no".	
<b>size</b> =["number(12)"]	Specifies the size of font used for the text of the center message. Default is "12".	
<b>color</b> =["Cstring(FFFFFF)"]	Specifies the color of font to be used for the text of the center message. Default is "FFFFFF" (white). Specify "FF0000" for red, "00FF00" for green, and "0000FF" for blue.	
<b>width</b> =["number(400)"]	Specifies the width of the text field in pixel used for the text of the center message. Default is "400".	
<b>height</b> =["number(40)"]	Specifies the height of the text field in pixel used for the text of the center message. Default is "100".	

### <hotspots\_enabled /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>hide</b> =["yes"] "no"]	Specifies if the message of "hotspots enable" is sidaplyed on the top-right corner of the pano-window. Default is "no".	all playmodes

## <play\_objects> tag

### <image /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>path</b> =["path/file_name"]	Specifies path and name or URL for the file including a still panorama image or a panoramic motion picture operable with the RyubinPanoPlayer. (note) If you specify the character string of "wait_for_selection" to this attribute for the video playmodes, the player stops its operation and waits for JavaScript API is called to select and load a video clip.	all playmodes
<b>cross_domain</b> =["yes" "no"]	Specifies if the image data specified by the <b>path</b> attribute resides in a different domain from the one where the player (swf file) resides. Even when you get approved from your Flash player to access the other domain by specifying cross_domain="yes", the actual access will be denied unless the target domain allows it via its security policy. Default setting is "no".	
<b>video_repeat</b> =["yes" "no"]	Specifies if the video is repeatedly played after reaching the end of the footage. Default setting is "yes".	video playmodes
<b>init_loudness</b> =["number(1.0)"] 0.0 <= value <= 1.0	Specifies the initial loudness of the sound when the player is launched. The values should be between 0.0 and 1.0. Default is "1.0".	video playmodes

### <sound /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>path</b> =["path/file_name"]	Specifies path and name or URL for the sound file created in mp3 format.	still playmodes
<b>cross_domain</b> =["yes" "no"]	Specifies if the sound file specified by the <b>path</b> attribute resides in different domain from the one where the player (swf file) resides. Even when you get approved from your Flash player to access the other domain by specifying cross_domain="yes", the actual access will be denied unless the target domain allows it via its security policy. Default setting is "no".	
<b>sound_repeat</b> =["yes" "no"]	Specifies if the sound is repeatedly played after reaching the end of the sound footage. Default setting is "yes".	still playmodes
<b>init_loudness</b> =["number(1.0)"] 0.0 <= value <= 1.0	Specifies the initial loudness of the sound when the player is launched. The values should be between 0.0 and 1.0. Default is "1.0".	still playmodes
<b>sound_clear</b> =["yes" "no"]	Specifies if the sound playing operation is cleared every time a sound file is loaded. Default setting is "yes". If you would like the player continuously play the current sound even when the new sound file is loaded (via JavaScript API), please specify "no".	still playmodes

<b>init_start</b> =["yes" "no"]	Specifies if the player starts playing a sound immediately after loading the sound file. Default is "yes". If you specify "no", the sound will not start playing until the play button is pressed.	still playmodes
---------------------------------	--	-----------------

## <crop /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>width</b> =["number(1.0)"] <b>height</b> =["number(1.0)"] <b>cx</b> =["number(0.5)"] <b>cy</b> =["number(0.5)"]	These attributes are used to determine an oval-shaped area to be cropped from the original image. The <b>cx</b> and <b>cy</b> are used to specify a relative coordinate of the center position of the oval. The <b>width</b> and <b>height</b> are used to specify horizontal and vertical diameters as relative values to the width and height of the original image, respectively. If specifying <b>cx</b> =0.5, <b>cy</b> =0.5, <b>width</b> =1.0 and <b>height</b> =1.0, the cropping area should be an oval (or circle) which is inscribed to the outer frame or the edges of the original image. The default value for <b>width</b> and <b>height</b> is 1.0. The default value for <b>cx</b> and <b>cy</b> is 0.5.	foughnut fisheye fish2rect verticalfisheye verticalfish2rect doughnutvideo fisheyevideo dualfisheyevideo fish2rectvideo verticalfisheyevideo verticalfish2rectvideo
<b>inner_circle</b> =["number(0.2)"] 0 < value < 0.5	Specifies the ratio of the inner radius to be cropped out, relative to the outer radius of the doughnut circle of the panorama image to be projected. Default value is 0.2.	doughnut doughnutvideo

## <projection /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>elevation</b> =["number(45)"] 0 < value <= 90	Specifies the angle of elevation in degree for the panorama image to be projected. Default is 45.	cylinder doughnut sphere cylindervideo doughnutvideo spherevideo
<b>depression</b> =["number(45)"] 0 < value <= 90	Specifies the angle of depression in degree for the panorama image to be projected. Default is 45.	
<b>horizontal_span</b> =["number(360)"] 0 < value <= 360	Specifies the angle of horizontal span in degree for the panorama image to be projected. Default is 360.	cylinder sphere cylindervideo spherevideo
<b>implicit</b> =["yes" "no"]	Specifies if the ratio of the height and with of the image is to be used for the actual projection (implicit="yes" ), or the attribute values of elevation and depression are to be used for the actual projection (implicit="no"). The default is implicit="yes".	sphere spherevideo
<b>proj_type</b> =["1" "2"]	Specifies if the orthogonal projection is applied to the doughnut image transformation. "1" for equi-distance projection, and "2" for orthogonal projection. Default is "1"; equi-distance projection.	doughnut doughnutvideo

## <view /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>init_fov</b> =["number(60)"] 20<=value<=130	Specifies the initial horizontal fov (field of view) in degree when the player is launched. Default is "60" for still playmodes and "100" for video playmodes. The value should be between "20" and "130"	all playmodes
<b>init_yaw</b> =["number(0)"] -180<= value<=180	Specifies the initial yaw in degree when the player is launched. Default is 0. The value should be between -180 and 180.	all playmodes
<b>init_pitch</b> =["number(0)"] -180<= value<=180	Specifies initial pitch in degree when the player is launched. Default is 0. The value should be between -180 and 180.	all playmodes
<b>init_view_mode</b> =["number(1)"]	Specifies the initial projection type when the player is launched. "1" for sphere mode (i.e. normal or rectilinear mode), "2" for globe mode (i.e. orthogonal mode), "3" for bipolar mode (i.e. hyperbolic or full-stereographic mode). Default is "1".	sphere sphere8 doughnut fisheye verticalfisheye spherevideo doughnutvideo fisheyevideo dualfisheyevideo verticalfisheyevideo
<b>init_fisheye_mode</b> =["number(1)"]	Specifies the initial rendering mode for the pay modes of fisheye and fisheyevideo when the player is launched. "1" for single mode, "2" for double mode, "3" for mirror mode. Default is "1".	fisheye fisheyevideo
<b>hv_ratio</b> =["number(2.0)"] 0.1<=value<=10	Specifies the aspect ratio (width/height) of the image when displayed. The value should be between 0.1 and 10.0. The default value are defined as follows: For the play modes of flat and flatvideo, the default is "2.0". For fish2rect and fish2rectvideo, the default is "1.0". For verticalfish2rect and verticalfish2rectvideo, the default is "4.0".	flat fish2rect verticalfish2rect doughnut2rectvideo flatvideo fish2rect verticalfish2rect
<b>init_filter</b> =["yes"/"no"]	Specifies if smoothing filter is enabled when the player is launched. Default is "no" (disabled).	all playmodes
<b>lock_nadirs</b> =["yes"/"no"]	Specifies if the viewing-direction is locked at the top and bottom nadirs. (i.e. the pitch movement will be locked at north and south poles; +90 degree and -90 degree). Default is "no" (unlocked).	all playmodes
<b>lock_vertical</b> =["yes"/"no"]	Specifies if viewing-direction is locked to the initial pitch value (i.e. pitch value will not be changed). Default is "no" (not locked).	all playmodes
<b>lock_fov</b> =["yes"/"no"]	Specifies if the fov or zooming operation is locked to the initial fov (i.e. fov value will not be changed). Default is "no" (unlocked).	all playmodes
<b>limit_vertical</b> =["yes"/"no"]	Specifies if the vertical travel or viewing-direction is limited within a span. Please note that the limit_vercial is forced to "no" when lock_nadirs="yes" or lock_vertical="yes" is specified. Default is "no" (not limited). When specified "yes" without using either the "top_limit" or "bottom_limit" below, the default setting	all playmodes

## Ryubin's Flash Panorama Player, User's Guide

	will be applied and the span will be from +90-degree to -90-degree; from top pole (zenith) to bottom pole (nadir).	
<b>top_limit</b> =["number(+90)"] +10<=value<=+90	Specifies the top limit for the span of vertical travel or viewing-direction in degree. Default value is +90. This attribute works only when limit_vertical="yes" is specified. Please note that the value of init_fov should be specified as smaller or equal to the value of top_limit plus bottom_limit, otherwise the viewing window may be blacked out.	all playmodes
<b>bottom_limit</b> =["number(+90)"] +10<=value<=+90	Specifies the bottom limit for the span of vertical travel or viewing-direction in degree. Default value is +90. This attribute works only when limit_vertical="yes" is specified. Please note that the value of init_fov should be specified as smaller or equal to the value of top_limit plus bottom_limit, otherwise the viewing window may be blacked out.	all playmodes
<b>limit_horizontal</b> =["yes" "no"]	Specifies if the horizontal travel or viewing-direction is limited within a span. Default is "no" (not limited). When specified "yes" without using either the "right_limit" or "left_limit" below, the default setting will be applied and the span will be from +90-degree to -90-degree.	all playmodes
<b>right_limit</b> =["number(+90)"] +10<=value<=+90	Specifies the right limit for the span of horizontal travel or viewing-direction in degree. Default value is +90. This attribute works only when limit_horizontal="yes" is specified.	all playmodes
<b>left_limit</b> =["number(+90)"] +10<=value<=+90	Specifies the left limit for the span of horizontal travel or viewing-direction in degree. Default value is +90. This attribute works only when limit_horizontal="yes" is specified.	all playmodes
<b>flip_vertical</b> =["yes" "no"]	Specifies if the image is to be vertically flipped, i.e, the image should be displayed upside down.	all playmodes
<b>flip_horizontal</b> =["yes" "no"]	Specifies if the image is to be horizontally flipped, i.e, the image should be displayed right-side left.	all playmodes
<b>seg_lock</b> =["high" "medium" "low" "no"]	This attribute is used to lock the "segment release" operation and to lock the number of segmented mesh. The character string of "high" gives the highest number and the "low" gives the lowest number for the segmentation. This attribute is intended to be a mean to compromise the quality and performance of the viewing experience. Please refer to <a href="#">"Balancing quality and performance"</a> for details.	doughnutVideo fisheyevideo dualfisheyevideo spherevideo
<b>quality_lock</b> =["high" "low" "no"]	This attribute is used to lock the "(rendering) quality release" operation and to lock the quality level of the rendering. The character string of "high" gives the higher rendering quality and the "low" gives the lower rendering quality. This attribute is intended to be a mean to compromise the quality and performance of the viewing experience. Please refer to <a href="#">"Balancing quality and performance"</a> for details.	doughnutvideo fisheyevideo dualfisheyevideo spherevideo
<b>disable_wheel</b> =["yes" "no"]	Specifies if the mouse wheel is disabled for zoom-in and zoom-out operation. Default is "no".	all playmodes
<b>disable_tailing</b> =["yes" "no"]	Specifies if the tailing behavior of the image movement is disabled. Default is "no".	all playmodes
<b>unlock_for_non_rectilinear</b> =["yes" "no"]	Specifies if the limitation of vertical travel (by limit_vertical attribute) is disabled when the projection type (view_mode) is	sphere sphere8

	"2" or "3". Default is "no".	doughnut fisheye verticalvisheye spherevideo doughnutvideo fisheyevideo dualfisheyevideo verticalfisheyevideo
--	------------------------------	--

### <auto /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>init_start</b> =["yes"/" <u>no</u> "]	Specifies if the auto-panning is to be started when the player is launched. Default is "no".	still playmodes
<b>pan_speed</b> =["number(-0.1)"] -20<= value <=20	Specifies the speed of auto-panning. The value is meant as degree per rendering-frame, which varies upon the actual operating environment. Default is -0.1.	
<b>tilt_amplitude</b> =["number(0)"] -180<= value <=180	Specifies the amplitude of vertical vibration in degree when auto-panning is working. Default is 0. The value should be between -180 and 180.	
<b>tilt_cycle</b> =["number(360)"] 10<= value <=3600	Specifies the cycle of vertical vibration in degree when auto-panning is working. Default is 360. The value should be between 10 and 3600.	

### <behavior /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>load_on_start</b> =["yes"/" <u>no</u> "]	Specifies if the video-load operation is to be started immediately at the first frame of the video clip. Default setting ="yes"	Video playmodes
<b>wait_for_preload</b> =["yes"/" <u>no</u> "]	Specifies if the video-play operation is to be suspended until the minimum pre-load condition is met. Default setting ="no"	
<b>pause_on_start</b> =["yes"/" <u>no</u> "]	Specifies if the video-play operation is to be started immediately at the first frame of the video clip. Default setting ="no"	
<b>start_button</b> =["yes"/" <u>no</u> "]	Specifies if the start button is to be made visible and activated. Default setting ="no"	
<b>preload_minimum</b> =["number(0.1)"] 0.0001< value <0.9999	Specifies the minimum preload condition. When wait_for_preload="yes", the video_play operation will pause until the minimum preload condition is met. The value represents the ratio of the preloaded video length per total video length.	
<b>preload_start</b> =["number(0.95)"] 0.90< value <0.99	Specifies the condition to start "video-preload" status, i.e. "preload_start" condition where "video_play" operation pauses to	



## Ryubin's Flash Panorama Player, User's Guide

	wait until "preload-stop" condition is met. The value represents the ratio of the already played video length (or video position, play head position) per the preloaded video length.	
<b>preload_stop</b> =["number(0.8)"] 0.60< value <89	Specifies the condition to top "video-preload" status, i.e. "preload_stop" condition where "video_play" operation resumes. The value represents the ratio of the already played video length (or video position, play head position) per the preloaded video length.	

### <info /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>text</b> =["Cstring"]	Specifies the text string used as a user definable information (UDI). The UDI can be shown / hidden by using a button that can be defined in the <a href="#">XML-based customizable controller</a> .	all playmodes
<b>font</b> =["Cstring(Arial Narrow)"]	Specifies the type of font such as "Arial" and "Arial Narrow" to be used for the UDI text. Default is "Arial Narrow"	all playmodes
<b>bold</b> =["yes"] "no"]	Specifies if the bold font is used for the UDI text. Default is "no".	
<b>italic</b> =["yes"] "no"]	Specifies if the italic font is used for the UDI text. Default is "no".	
<b>size</b> =["number(12)"]	Specifies the size of font used for the UDI text. Default is "12".	
<b>color</b> =["Cstring(FFFFFF)"]	Specifies the color of font to be used for the UDI text. Default is "FFFFFF" (white). Specify "FF0000" for red, "00FF00" for green, and "0000FF" for blue.	
<b>bg</b> =["yes"] "no"]	Specifies if the background is used for the UDI text. Default is "no".	
<b>bg_color</b> =["Cstring(FFFFFF)"]	Specifies the color of the background to be used for the UDI text. Default is "000000" (black). Specify "FF0000" for red, "00FF00" for green, and "0000FF" for blue.	
<b>width</b> =["number(400)"]	Specifies the width of the text field in pixel used for the UDI text. Default is "400".	
<b>height</b> =["number(100)"]	Specifies the height of the text field in pixel used for the UDI text. Default is "100".	
<b>align</b> =["left"] "center"] "right"]	Specifies if the alignment of the UDI text within the text field. Default is "left".	
<b>alignx</b> =["left"] "center"] "right"] <b>aligny</b> =["top"] "center"] "bottom"]	Specifies the original position of the UDI text field by means of 3x3 alignment values within the pano-window. Defaults are "center" for <b>alignx</b> (horizontal alignment) and "center" for <b>aligny</b> (vertical alignment) as well.	
<b>posx</b> =["number(0)"]	Specifies the relative position (or shift values in pixel) of the UDI text field to the original position. Default is "0" both for <b>posx</b> (horizontal shift) and	

## Ryubin's Flash Panorama Player, User's Guide

<b>posy</b> =["number(0)"]	<b>posy</b> (vertical shift).	
-------------------------------	-------------------------------	--

### <caps /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>path_zenith</b> =["path/file_name"]	Specifies path and name or URL for the image file used as a zenith cap. The image file should be in a png or jpg format. The zenith cap is an image data which is intended to overlay or hide a black hole usually shown at the north or upper pole in the play modes of doughnut or doughnut video. Default setting is "cap_zenith.png".	doughnut doughnutvideo spherevideo
<b>path_nadir</b> =["path/file_name"]	Specifies path and name or URL for the image file used as a nadir cap. The image file should be in png or jpg format. The nadir cap is an image data which is intended to overlay or hide a black hole usually shown at the south or lower pole in the play modes of doughnut or doughnut video as well as spherevideo. Default setting is "cap_nadir.png".	
<b>roll_zenith</b> =["number(0)"]	Specifies the value of roll effect or rotation in degree you would like to apply on the zenith cap. Default is "0".	
<b>roll_nadir</b> =["number(0)"]	Specifies the value of roll effect or rotation in degree you would like to apply on the nadir cap. Default is "0".	

### <hot\_spots> tag

#### <add /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>yaw</b> =["number(0)"] -180< value <180	Specifies the yaw value of the designated hot spot, in degree. Default setting is 0.	All playmodes
<b>pitch</b> =["number(0)"] -180< value <180	Specifies the pitch value of the designated hot spot, in degree. Default setting is 0.	
<b>img_path</b> =["path/file_name"]	Specifies path and name or URL for the hot spot image created in jpg, png or gif format. Default setting is RPN + "png".	
<b>caption</b> =["Cstring"]	Specifies a character string which is made visible when the mouse cursor moves over the hot spot. Default setting is "" (blank).	

## Ryubin's Flash Panorama Player, User's Guide

<b>action</b> =["roam" "warp" "link"]	Specifies the type of action when the designated hot spot is clicked. Default setting is "link"	
<b>dest</b> =["path/file_name" "URL"]	If <b>action</b> ="roam" or <b>action</b> ="warp", the <b>dest</b> attribute specifies path and name or URL for the XML file that defines the scene to be displayed when the designated hot spot is clicked. If <b>action</b> ="link", the <b>dest</b> attribute specifies the URL to an external or internal Web page to be displayed when the designated hot spot is clicked.	
<b>target</b> =["_blank" "_parent" "_self" "_top"]	Specifies the name of the window where the linked Web page is displayed. Default setting is "_blank". This attribute works only when <b>action</b> ="link", otherwise this is neglected.	

### <play\_list> tag

#### <add /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>item</b> =["path/file_name"]	Specifies path and name or URL of the xml file to be added to the play list. The xml file should include the definitions and the behaviors of the video to be played. The player endlessly plays the videos according to the sequence of the item. When reaching the end of the list, it returns to the top of the list.	video playmodes

### <user\_loader> tag

#### <caption /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>path</b> =["path/file_name"]	Specifies path and name or URL for the caption file, which is an external image file to be used in place of the text of "loading" which blinks while the panorama image is being loaded.	still playmodes

<b>posx</b> =["number(0)"] <b>posy</b> =["number(0)"]	Both attributes, to be set as pixel values, in combination specify the location in the stage where the caption image is intended to be placed. When both attributes are not specified or set to zeros, the center of the caption image will be positioned to the center of the stage, meaning that <b>posx</b> and <b>posy</b> are used to shift the location of the caption image relatively to this original position. The default value for both attributes is zero.	
--	---	--

## <indicator /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>path</b> =["path/file_name"]	Specifies path and name or URL for the indicator file, which is an external image file to be used in place of the progress bar whose length (horizontal size or width) varies in sync with the progress of the image loading operation.	still playmodes
<b>posx</b> =["number(0)"] <b>posy</b> =["number(0)"]	Both attributes, to be set as pixel values, in combination specify the location in the stage where the indicator image is intended to be placed. When both attributes are not specified or set to zeros, the center of the indicator image will be positioned to the center of the stage, meaning that <b>posx</b> and <b>posy</b> are used to shift the location of the indicator image relatively to this original position. The default value for both attributes is zero.	

## <background /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>path</b> =["path/file_name"]	Specifies path and name or URL for the background file, which is an external image file to be used as a background image for the progress indicator while the panorama image is being loaded.	still playmodes
<b>posx</b> =["number(0)"] <b>posy</b> =["number(0)"]	Both attributes, to be set as pixel values, in combination specify the location in the stage where the background image is intended to be placed. When both attributes are not specified or set to zeros, the center of the background image will be positioned to the center of the stage, meaning that <b>posx</b> and <b>posy</b> are used to shift the location of the background image relatively to this original position. The default value for both attributes is zero.	

## <user\_panel> tag

## <prologue /> element

ATTRIBUTE	FUNCTION	APPLICABLE
-----------	----------	------------

## Ryubin's Flash Panorama Player, User's Guide

		PLAYMODE(S)
<b>path</b> =["path/file_name"]	Specifies path and name or URL for the prologue file, which is an external image file to be used as a prologue image to be displayed while the panorama image is being loaded.	still playmodes
<b>posx</b> =["number(0)"] <b>posy</b> =["number(0)"]	Both attributes, to be set as pixel values, in combination specify the location in the stage where the prologue image is intended to be placed. When both attributes are not specified or set to zeros, the center of the prologue image will be positioned to the center of the stage, meaning that <b>posx</b> and <b>posy</b> are used to shift the location of the prologue image relatively to this original position. The default value for both attributes is zero.	

### <u\_context1 /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>path</b> =["path/file_name"]	Specifies path and name or URL for a user-definable image to be displayed after the panorama image is loaded.	all playmodes
<b>posx</b> =["number(0)"] <b>posy</b> =["number(0)"]	Both attributes, to be set as pixel values, in combination specify the location in the stage where the user-definable image is intended to be placed. When both attributes are not specified or set to zeros, the center of the user-definable image will be positioned to the center of the stage, meaning that <b>posx</b> and <b>posy</b> are used to shift the location of the user-definable image relatively to this original position. The default value for both attributes is zero.	

### <u\_context2 /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>path</b> =["path/file_name"]	Specifies path and name or URL for a user-definable image to be displayed after the panorama image is loaded.	all playmodes
<b>posx</b> =["number(0)"] <b>posy</b> =["number(0)"]	Both attributes, to be set as pixel values, in combination specify the location in the stage where the user-definable image is intended to be placed. When both attributes are not specified or set to zeros, the center of the user-definable image will be positioned to the center of the stage, meaning that <b>posx</b> and <b>posy</b> are used to shift the location of the user-definable image relatively to this original position. The default value for both attributes is zero.	

### <skin /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
-----------	----------	------------------------

<b>path</b> =["path/file_name"]	Specifies path and name or URL for a skin which is a user-definable image to be used as a background for the stage.	all playmodes
<b>posx</b> =["number(0)"] <b>posy</b> =["number(0)"]	Both attributes, to be set as pixel values, in combination specify the location in the stage where the skin image is intended to be placed. When both attributes are not specified or set to zeros, the top-left corner of the skin image will be positioned to the top-left corner of the stage, meaning that <b>posx</b> and <b>posy</b> are used to shift the location of the skin relatively to this original position. The default value for both attributes is zero.	

## <logo /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>path</b> =["path/file_name"]	Specifies path and name or URL for a logo image to be used.	all playmodes
<b>posx</b> =["number(0)"] <b>posy</b> =["number(0)"]	Both attributes, to be set as pixel values, in combination specify the relative location in the pano-window (or the stage) where the logo image is intended to be placed. When both attributes are not specified or set to zeros, the top-left corner of the logo image will be positioned to the top-left corner of the pano-window as default (see the <b>anchor</b> attribute below), meaning that <b>posx</b> and <b>posy</b> are used to shift the location of the logo image relative to the original position. The default value for both attributes is zero.	
<b>anchor</b> =["tl" "tr" "bl" "br"]	Specifies the corner of the pano-window (or the stage when the <b>use_pw</b> ="no" as described below.) to which the logo image should be anchored. "tl" for top-left, "tr" for top-right, "bl" for bottom-left and "br" for bottom-right. Default is "tl".	all playmodes
<b>use_pw</b> =["no" "yes"]	If you would like anchor the logo image to one of the corners of the stage instead of the pano-window, specify use_pw="no". Default is "yes" (use pano-window to anchor the logo image).	

## <context\_menu> tag

### <u\_context1 /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	all playmodes
<b>on_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is disabled. The default character string is "Show My Statement".	

<b>off_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is enabled. The default character string is " <a href="#">Hide My Statement</a> ".	
---------------------------------	--	--

## <u\_context2 /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	all playmodes
<b>on_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is disabled. The default character string is " <a href="#">Show My Message</a> ".	
<b>off_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is enabled. The default character string is " <a href="#">Hide My Message</a> ".	

## <home\_pos /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	all playmodes
<b>caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item. The default character string is " <a href="#">Home Positon</a> ".	

## <show\_info /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	all playmodes
<b>on_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is disabled. The default character string is " <a href="#">Show View Info</a> ".	
<b>off_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is enabled. The default character string is " <a href="#">Hide View Info</a> ".	

### <smoothing /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	all playmodes
<b>on_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is disabled. The default character string is "Enable Smooth Filter".	
<b>off_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is enabled. The default character string is "Disable Smooth Filter".	

### <panorama\_controller /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	all playmodes
<b>on_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is disabled. The default character string is "Show Panorama Controller".	
<b>off_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is enabled. The default character string is "Hide Panorama Controller".	

### <hot\_spots /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	cube cube6 cube24 flat sphere sphere8 cubevideo flatvideo spherevideo
<b>on_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is disabled. The default character string is "Enable Hotspots".	
<b>off_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is enabled. The default character string is "Disable Hotspots".	



### <sound\_controller /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	still playmodes
<b>on_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is disabled. The default character string is " <a href="#">Show Sound Controller</a> ".	
<b>off_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is enabled. The default character string is " <a href="#">Hide Sound Controller</a> ".	

### <video\_controller /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	video playmodes
<b>on_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is disabled. The default character string is " <a href="#">Show Video Controller</a> ".	
<b>off_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is enabled. The default character string is " <a href="#">Hide Video Controller</a> ".	

### <start\_auto /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	still playmodes
<b>on_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is disabled. The default character string is " <a href="#">Start Auto Panning</a> ".	
<b>off_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is enabled. The default character string is " <a href="#">Stop Auto Panning</a> ".	

## <full\_screen /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	all playmodes
<b>on_caption</b> ["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is disabled. The default character string is "Enable Fullscreen".	
<b>off_caption</b> ["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant function is enabled. The default character string is "Disable Fullscreen".	

## <about\_me /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context menu. Default setting is "yes".	all playmodes
<b>my_name</b> ["Cstring"]	Specifies a character string you would like to show in the context menu as your brand, company name or else. Default setting is "(c) 20xx Ryubin's Panorama Laboratory".	
<b>my_link</b> ["Cstring"]	Specifies a URL to the link you would like to open when the above character string is clicked on. Default setting is " <a href="http://ryubinpanorama.com">http://ryubinpanorama.com</a> ".	

## <sphere\_context> tag

### <normal /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	sphere sphere8 doughnut fisheye verticalvisheye spherevideo doughnutvideo fisheyevideo dualfisheyevideo verticalfisheyevideo
<b>on_caption</b> ["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant mode is selected. The default character string is "--> Normal Mode".	
<b>off_caption</b> ["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant mode is not selected. The default character string is	

	"Normal Mode".	
--	----------------	--

### <globe /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	sphere sphere8 doughnut fisheye verticalvisheye spherevideo doughnutvideo fisheyevideo dualfisheyevideo verticalfisheyevideo
<b>on_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant mode is selected. The default character string is "--> <a href="#">Globe Mode</a> ".	
<b>off_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant mode is not selected. The default character string is " <a href="#">Globe Mode</a> ".	

### <bipolar /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	sphere sphere8 doughnut fisheye verticalvisheye spherevideo doughnutvideo fisheyevideo dualfisheyevideo verticalfisheyevideo
<b>on_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant mode is selected. The default character string is "--> <a href="#">Bipolar Mode</a> ".	
<b>off_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant mode is not selected. The default character string is " <a href="#">Bipolar Mode</a> ".	

### <fisheye\_context> tag

#### <single /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	fisheye fisheyevideo

## Ryubin's Flash Panorama Player, User's Guide

<b>on_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant mode is selected. The default character string is "--> Single Mode".	
<b>off_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant mode is not selected. The default character string is "Single Mode".	

### <double /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	fisheye fisheyevideo
<b>on_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant mode is selected. The default character string is "--> Double Mode".	
<b>off_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant mode is not selected. The default character string is "Double Mode".	

### <mirror /> element

ATTRIBUTE	FUNCTION	APPLICABLE PLAYMODE(S)
<b>show_item</b> =["yes" "no"]	Specifies whether or not this menu item is included in the context_menu. Default setting is "no".	fisheye fisheyevideo
<b>on_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant mode is selected. The default character string is "--> Mirror Mode".	
<b>off_caption</b> =["Cstring"]	Specifies a character string to be shown in the position of the menu item while the relevant mode is not selected. The default character string is "Mirror Mode".	

## Specifications of the XML-based customizable controller

### Format

Here is a complete format of the XML-based customizable controller:

```
<?xml version = '1.0'?>
<any name>
  <object>
    <category>Cstring</category>
    <caption>Cstring</caption>
    <adjust>Cstring</adjust>
    <pos_x>signed integer</pos_x>
    <pos_y>signed integer</pos_y>
    <path>path/file_name</path>
    <up_path>path/file_name</up_path>
    <over_path>path/file_name</over_path>
    <down_path>path/file_name</down_path>
    <width>unsigned integer</width>
    <action>Cstring</action>
    <role>Cstring</role>
  </object>
</any name >
```

### General rule

Creating an XML-based customizable controller for the RyubinPanoPlayer is a process of designing the images and layouts of multiple images, buttons and text fields under a pre-defined rule as follows:

- 1) The highest level tag can have any name such as `<any_name></any_name>` .

- 2) The second level tag should be a pair of <object> and </object> tags.
- 3) The third level tags which are placed between the <object> and </object> tags define the characteristics of each object.
- 4) The objects will be laid in the first-under latter-over manner, meaning that a new object will be laid over the previous one according to the sequence of the pairs of <object> and </object> tags.
- 5) The number of objects should be equal to or less than 40 in a single XML file.
- 6) The objects are classified into three categories, "image", "text" and "button".
- 7) The category should be specified by the <category></category> tags. If no <category> tag is specified, it is meant to be classified as the button category, as default.
- 8) Each category has different set of attributes that should be specified by the user when creating the XML file as follows:

Attribute (The third level tag)	Category		
	image	text	button
<category>	"image" (mandatory)	"text"(mandatory)	"button"(optional)
<caption>	n/a	"Cstring"(mandatory)	"Cstring"(optional)
<posx>	number of pixel as signed integer (mandatory)	number of pixel as signed integer (optional)	number of pixel as signed integer (mandatory)
<posy>	number of pixel as signed integer (mandatory)	number of pixel as signed integer (mandatory)	number of pixel as signed integer (mandatory)
<path>	path/name of the image file (mandatory)	n/a	n/a
<up_path>	n/a	n/a	path/name of the image file (mandatory)
<over_path>	n/a	n/a	path/name of the image file (mandatory)
<down_path>	n/a	n/a	path/name of the image file (mandatory)
<width>	number of pixel as unsigned integer (optional)	n/a	n/a
<action>	n/a	n/a	See <a href="#">&lt;action table&gt;</a> below. (mandatory)
<role>	See <a href="#">&lt;role table-A&gt;</a> below (optional)	See <a href="#">&lt;role table-B&gt;</a> below. (mandatory)	n/a

#### <attribute table>

Attribute (The third level tag)	Description
<category>	Specifies the character string to classify the category of the object: "image", "text" or "button". If omitted, the object will be classified to the "button" category.
<caption>	This attribute is applicable to the objects in the category of "button" and "text". For a button object, this attribute optionally specifies a character string supposed to be used as a message to explain the usage of the button, and the message will be shown

	when the mouse cursor is placed over the button image. For a text object, this is merely used as a text string of the text object.
<posx>	Specifies the x-coordinate of the object relative to the left edge of the controller. It's recommended to set 0 to this attribute for the first object in your XML code.
<posy>	Specifies the y-coordinate of the object. The y-coordinate is handled as a non-absolute or relative positional value which means that only the differences of the y-coordinates among multiple objects are concerned when making a layout of the controller. It's recommended to set 0 to this attribute for the first object in your XML code.
<path>	Specifies the path/name of the image file used for the image object.
<up_path>	Specifies the path/name of the image file used for the "up image" of the button object. The "up image" is the face image of the button when it's in the normal state.
<over_path>	Specifies the path/name of the image file used for the "over image" of the button object. The "over image" is the face image of the button when the mouse cursor or pointer is placed on the button object.
<down_path>	Specifies the path/name of the image file used for the "down image" of the button object. The "down image" is the face image of the button when the button is pressed.
<width>	Specifies the width of the image object by a number of pixels (optional). When no number is specified, the actual width of the image data will be used as the width of the image object.
<action>	Specifies the character string for the action assigned to this "button" object. See <a href="#">&lt;action table&gt;</a> for details.
<role>	Specifies the character string for the role assigned to this "image" or "text" object. See <a href="#">&lt;action table&gt;</a> for details. See <a href="#">&lt;role table-A&gt;</a> and <a href="#">&lt;role table-B&gt;</a> below for details.

- 9) The action attribute (the third level tag) solely used by the button object are defined in the <action table> below:

<action table> used for button objects

<i>Cstring</i> for action	function
plus	While the button associated to this action is being pressed, the player zooms in the image displayed in the pano-window (i.e. decreases the field of view).
minus	While the button associated to this action is being pressed, the player zooms out the image displayed in the pano-window (i.e. increases the field of view).
right	While the button associated to this action is being pressed, the player moves the viewing direction rightward in the pano-window (i.e. moves the image leftward).
left	While the button associated to this action is being pressed, the player moves the viewing direction leftward in the pano-window (i.e. moves the image rightward).
up	While the button associated to this action is being pressed, the player moves the viewing direction upward in the pano-window (i.e. moves the image downward).
down	While the button associated to this action is being pressed, the player moves the viewing direction downward in the pano-window (i.e. moves the image upward).
d_right	When the button associated to this action is pressed, the player moves the viewing direction by 90-degree rightward.
d_left	When the button associated to this action is pressed, the player moves the viewing direction by 90-degree leftward.
home	When the button associated to this action is pressed, the player returns the viewing direction to the home position.
play	When the button associated to this action is pressed, the player starts playing the designated video or the sound. The play button will be disabled and invisible while playing the video or the sound.
pause	When the button associated to this action is pressed, the player pauses playing the designated video or the sound. The pause button will be disabled and invisible while pausing.

<a href="#">rewind</a>	When the button associated to this action is pressed, the players goes back to the beginning position of the designated video or the sound, and resume playing.
<a href="#">turn_on</a>	When the button associated to this action is pressed, the players turn off the sound if the sound exists. The <a href="#">turn_on</a> button will be disabled and invisible while the sound is turned on.
<a href="#">turn_off</a>	When the button associated to this action is pressed, the players turn on the sound if the sound exists. The <a href="#">turn_off</a> button will be disabled and invisible while the sound is turned off.
<a href="#">full_scr</a>	When the button associated to this action is pressed, the player toggles the screen mode between the full screen mode and the normal mode.
<a href="#">auto_start</a>	When the button associated to this action is pressed, the player starts the auto panning operation. The <a href="#">auto_start</a> button will be disabled and invisible while the auto-pan is running.
<a href="#">auto_stop</a>	When the button associated to this action is pressed, the player stops the auto panning operation. The <a href="#">auto_stop</a> button will be disabled and invisible while the auto-pan is not running.
<a href="#">projection0</a>	When the button associated to this action is pressed, the player changes the projection into rectilinear mode.
<a href="#">projection1</a>	When the button associated to this action is pressed, the player changes the projection into semi stereographic mode.
<a href="#">projection2</a>	When the button associated to this action is pressed, the player changes the projection into stereographic mode.
<a href="#">sound_effect0</a>	When the button associated to this action is pressed, the player changes the sound effect mode onto normal.
<a href="#">sound_effect1</a>	When the button associated to this action is pressed, the player changes the sound effect into "stereo rotation mode", where the sound of both right channel and the left channel rotates at the same time according to the yaw value. When the yaw value reaches 180-degree, the right channel and the left channel will be swapped.
<a href="#">sound_effect2</a>	When the button associated to this action is pressed, the player changes the sound effect into "monaural rotation mode", where the sound of right channel and the left channel are mixed into a monaural and rotates according to the yaw value. When the yaw value reaches 180-degree (or opposite to the home position), the loudness of the monaural sound becomes to its minimum. When the yaw value is 0 (or at the home position), the loudness of the monaural sound becomes its maximum.
<a href="#">view_experience</a>	When the button associated to this action is pressed, the player shows/hides the system variables such as yaw, pitch, fov and others at the top-left corner of the pano-window.

<role table-A> used for image objects

<i>Cstring</i> for role	function
<a href="#">prog_loaded</a>	The image assigned this role works as a progress bar that shows the amount of already loaded video file or sound file by its width.
<a href="#">prog_played</a>	The image assigned this role works as a progress bar that shows the amount of already played video file or sound file by its width. In other words, this image is intended to show the current play position. This image is supposed to be laid over the <a href="#">prog_loaded</a> image.
<a href="#">prog_back</a>	The image assigned this role works as a background image supposed to be laid under the <a href="#">prog_loaded</a> image as well as <a href="#">prog_played</a> image. The width of this image is used to measure the amount of the loaded as well as the played amount of the video or the sound. When the width of the <a href="#">prog_loaded</a> or the <a href="#">prog_played</a> image becomes equal to the width of the <a href="#">prog_back</a> , it means that 100% of the video or the sound is already loaded or played.
<a href="#">volume_pos</a>	The image assigned this role works as a slide bar that shows the loudness of the sound by its width.
<a href="#">volume_back</a>	The image assigned this role works as a background image supposed to be laid under the <a href="#">volume_pos</a> image. The width of this image is used to measure the loudness of the sound. When the width of the <a href="#">volume_pos</a> image becomes equal to the width of the <a href="#">volume_back</a> , it means that the loudness of the sound is at its maximum (100%).



<role table-B> used for text objects

<i>Cstring</i> for role	function
<code>prog_indicator</code>	The text assigned this role works as a progress indicator that shows the current play position of the video or the sound in a format of "ss:mm / SS:MM", where "ss:mm" is the current play position, "SS:MM" shows the total length of the video or the sound, "ss" and "SS" represent two-digit values of the second and mm and MM represent two-digit values of the minute.
<code>pos_indicator</code>	The text assigned this role works as a positional indicator. This text will be made visible while the mouse pointer is located over the <code>prog_loaded</code> image or the <code>prog_played</code> image, and changes its horizontal position( x-coordinate) according to the horizontal position (x-coordinate) of the mouse pointer. The vertical position (y-coordinate) of the text will be unchanged and locked to the value defined by <code>pos_y</code> attribute of the <code>pos_indicator</code> text. The text will be shown in a format of "ss:mm" and represents the value of assumed play position where the mouse pointer is located.
<code>volume_indicator</code>	The text assigned this role works as a positional indicator. This text will be made visible while the mouse pointer is located over the <code>volume_back</code> image or the <code>volume_pos</code> image, and changes its horizontal position( x-coordinate) according to the horizontal position (x-coordinate) of the mouse pointer. The vertical position (y-coordinate) of the text will be unchanged and locked to the value defined by <code>pos_y</code> attribute of the <code>volume_indicator</code> text. The text will be shown in a format of "n.n" (from 0.0 to 1.0) and represents the value of assumed loudness where the mouse pointer is located. The number 1.0 indicates the loudness of the sound is at its maximum.

## Basic design concept

The XML based customizable controller can be independent from the size (width and height) of the panorama window or the stage. That means that <posx> and <posy> within the <object> tag is a kind of positioning values relative to the origin (0,0) which you virtually or conceptually define at the top-left corner of the layout.

Therefore, once you have created a layout or a design, you do not need to change the XML code of the controller even if you change the size of the screen. The controller can be aligned or placed at one of the nine locations according to the combination of `alignx=""` and `aligny=""` attributes in the <panorama\_controller>, the <sound\_controller> or the <video\_controller> element.

The string of `alignx` can be left, center or right. The string of `aligny` can be top, center or bottom. Precise positioning can be carried out by using `posx` and `posy` attributes of the <panorama\_controller> or the <sound\_controller> element.

On the bottom line, the XML based customizable controllers are reusable for whatever design or layout of the panowindow or the stage.

## Sample code 1 (IntegratedCtrl.xml)

```
<?xml version = '1.0'?>
<video_controller>
  <object>
    <category>image</category>
    <pos_x>0</pos_x>
    <pos_y>0</pos_y>
    <path>object_images/ctrl_back01.png</path>
    <width>500</width>
  </object>
  <object>
    <category>button</category>
    <caption>Rewind & Play</caption>
    <adjust>left</adjust>
    <pos_x>0</pos_x>
    <pos_y>0</pos_y>
    <up_path>object_images/rewind01.png</up_path>
    <over_path>object_images/rewind02.png</over_path>
    <down_path>object_images/rewind03.png</down_path>
    <action>rewind</action>
  </object>
  <object>
    <category>button</category>
    <caption>Play</caption>
    <pos_x>25</pos_x>
    <pos_y>0</pos_y>
    <up_path>object_images/play01.png</up_path>
    <over_path>object_images/play02.png</over_path>
    <down_path>object_images/play03.png</down_path>
    <action>play</action>
  </object>
  <object>
    <category>button</category>
    <caption>Pause</caption>
    <pos_x>25</pos_x>
    <pos_y>0</pos_y>
    <up_path>object_images/pause01.png</up_path>
    <over_path>object_images/pause02.png</over_path>
    <down_path>object_images/pause03.png</down_path>
    <action>pause</action>
  </object>
  <object>
    <category>image</category>
    <pos_x>55</pos_x>
    <pos_y>0</pos_y>
    <path>object_images/prog_back01.png</path>
    <width>170</width>
    <role>prog_back</role>
  </object>
  <object>
    <category>image</category>
    <pos_x>55</pos_x>
    <pos_y>0</pos_y>
    <path>object_images/prog_loaded01.png</path>
    <width>170</width>
    <role>prog_loaded</role>
  </object>
  <object>
    <category>image</category>
    <pos_x>55</pos_x>
    <pos_y>0</pos_y>
    <path>object_images/prog_played01.png</path>
    <width>170</width>
    <role>prog_played</role>
  </object>
  <object>
    <category>text</category>
    <pos_x>225</pos_x>
    <pos_y>3</pos_y>
    <caption>00.00 / 00.00</caption>
    <width>80</width>
    <height>20</height>
    <role>prog_indicator</role>
  </object>
  <object>
    <category>text</category>
    <pos_x></pos_x>
    <pos_y>19</pos_y>
    <caption>00.00</caption>
    <width>40</width>
    <height>18</height>
    <role>pos_indicator</role>
  </object>
  <object>
    <category>image</category>
    <pos_x>330</pos_x>
    <pos_y>0</pos_y>
    <path>object_images/prog_back01.png</path>
    <width>60</width>
    <role>volume_back</role>
  </object>
</video_controller>
```

## Ryubin's Flash Panorama Player, User's Guide

```
<object>
    <category>image</category>
    <pos_x>330</pos_x>
    <pos_y>0</pos_y>
    <path>object_images/prog_played01.png</path>
    <width>60</width>
    <role>volume_pos</role>
</object>
<object>
    <category>text</category>
    <pos_x></pos_x>
    <pos_y>19</pos_y>
    <caption>X.X</caption>
    <width>30</width>
    <height>18</height>
    <role>volume_indicator</role>
</object>
<object>
    <category>button</category>
    <caption>Turn On Sound</caption>
    <pos_x>300</pos_x>
    <pos_y>0</pos_y>
    <up_path>object_images/sound_on01.png</up_path>
    <over_path>object_images/sound_on02.png</over_path>
    <down_path>object_images/sound_on03.png</down_path>
    <action>turn_on</action>
</object>
<object>
    <category>button</category>
    <caption>Turn Off Sound</caption>
    <pos_x>300</pos_x>
    <pos_y>0</pos_y>
    <up_path>object_images/sound_off01.png</up_path>
    <over_path>object_images/sound_off02.png</over_path>
    <down_path>object_images/sound_off03.png</down_path>
    <action>turn_off</action>
</object>
<object>
    <category>button</category>
    <caption>Show / Hide Information</caption>
    <pos_x>400</pos_x>
    <pos_y>0</pos_y>
    <up_path>object_images/info01.png</up_path>
    <over_path>object_images/info02.png</over_path>
    <down_path>object_images/info03.png</down_path>
    <action>show_information</action>
</object>
<object>
    <category>button</category>
    <caption>Zoom Out</caption>
    <pos_x>425</pos_x>
    <pos_y>0</pos_y>
    <up_path>object_images/minus01.png</up_path>
    <over_path>object_images/minus02.png</over_path>
    <down_path>object_images/minus03.png</down_path>
    <action>minus</action>
</object>
<object>
    <category>button</category>
    <caption>Zoom In</caption>
    <pos_x>450</pos_x>
    <pos_y>0</pos_y>
    <up_path>object_images/plus01.png</up_path>
    <over_path>object_images/plus02.png</over_path>
    <down_path>object_images/plus03.png</down_path>
    <action>plus</action>
</object>
<object>
    <category>button</category>
    <caption>Enter / Exit Fullscreen Mode</caption>
    <pos_x>475</pos_x>
    <pos_y>0</pos_y>
    <adjust>right</adjust>
    <up_path>object_images/display01.png</up_path>
    <over_path>object_images/display02.png</over_path>
    <down_path>object_images/display03.png</down_path>
    <action>full_scr</action>
</object>
</video_controller>
```

## Sample code 2 (ArrowButtons.xml)

```
<?xml version = '1.0'?>
<panorama_controller>
  <object>
    <category>image</category>
    <pos_x>0</pos_x>
    <pos_y>0</pos_y>
    <path>object_images/pano_ctrl_back01.png</path>
  </object>
  <object>
    <pos_x>25</pos_x>
    <pos_y>0</pos_y>
    <!-- caption>View Up</caption -->
    <up_path>object_images/up01.png</up_path>
    <over_path>object_images/up02.png</over_path>
    <down_path>object_images/up03.png</down_path>
    <action>up</action>
  </object>
  <object>
    <pos_x>0</pos_x>
    <pos_y>25</pos_y>
    <!-- caption>View Left</caption -->
    <up_path>object_images/left01.png</up_path>
    <over_path>object_images/left02.png</over_path>
    <down_path>object_images/left03.png</down_path>
    <action>left</action>
  </object>
  <object>
    <pos_x>25</pos_x>
    <pos_y>25</pos_y>
    <!-- caption>Home</caption -->
    <up_path>object_images/home01.png</up_path>
    <over_path>object_images/home02.png</over_path>
    <down_path>object_images/home03.png</down_path>
    <action>home</action>
  </object>
  <object>
    <pos_x>50</pos_x>
    <pos_y>25</pos_y>
    <!-- caption>View Right</caption -->
    <adjust>right</adjust>
    <up_path>object_images/right01.png</up_path>
    <over_path>object_images/right02.png</over_path>
    <down_path>object_images/right03.png</down_path>
    <action>right</action>
  </object>
  <object>
    <pos_x>25</pos_x>
    <pos_y>50</pos_y>
    <!-- caption>View Down</caption -->
    <up_path>object_images/down01.png</up_path>
    <over_path>object_images/down02.png</over_path>
    <down_path>object_images/down03.png</down_path>
    <action>down</action>
  </object>
  <object>
    <pos_x>0</pos_x>
    <pos_y>75</pos_y>
    <!-- caption>Pan Left 90-deg</caption -->
    <up_path>object_images/fast_left01.png</up_path>
    <over_path>object_images/fast_left02.png</over_path>
    <down_path>object_images/fast_left03.png</down_path>
    <action>d_left</action>
  </object>
  <object>
    <pos_x>50</pos_x>
    <pos_y>75</pos_y>
    <!-- caption>Pan Right 90-deg</caption -->
    <adjust>right</adjust>
    <up_path>object_images/fast_right01.png</up_path>
    <over_path>object_images/fast_right02.png</over_path>
    <down_path>object_images/fast_right03.png</down_path>
    <action>d_right</action>
  </object>
  <object>
    <pos_x>25</pos_x>
    <pos_y>75</pos_y>
    <!-- caption>Start Auto Pan</caption -->
    <adjust>right</adjust>
    <up_path>object_images/auto_start01.png</up_path>
    <over_path>object_images/auto_start02.png</over_path>
    <down_path>object_images/auto_start03.png</down_path>
    <action>auto_start</action>
  </object>
  <object>
    <pos_x>25</pos_x>
    <pos_y>75</pos_y>
    <!-- caption>Stop Auto Pan</caption -->
    <adjust>right</adjust>
```

## Ryubin's Flash Panorama Player, User's Guide

```
<up_path>object_images/auto_stop01.png</up_path>
<over_path>object_images/auto_stop02.png</over_path>
<down_path>object_images/auto_stop03.png</down_path>
<action>auto_stop</action>
</object>
<object>
  <pos_x>0</pos_x>
  <pos_y>125</pos_y>
  <!-- caption>Normal Suond</caption -->
  <up_path>object_images/s001.png</up_path>
  <over_path>object_images/s002.png</over_path>
  <down_path>object_images/s003.png</down_path>
  <action>sound_effect0</action>
</object>
<object>
  <pos_x>25</pos_x>
  <pos_y>125</pos_y>
  <!-- caption>Sound Effect 1</caption -->
  <adjust>right</adjust>
  <up_path>object_images/s101.png</up_path>
  <over_path>object_images/s102.png</over_path>
  <down_path>object_images/s103.png</down_path>
  <action>sound_effect1</action>
</object>
<object>
  <pos_x>50</pos_x>
  <pos_y>125</pos_y>
  <!-- caption>Sound Effect 2</caption -->
  <adjust>right</adjust>
  <up_path>object_images/s201.png</up_path>
  <over_path>object_images/s202.png</over_path>
  <down_path>object_images/s203.png</down_path>
  <action>sound_effect2</action>
</object>
<object>
  <pos_x>0</pos_x>
  <pos_y>150</pos_y>
  <!-- caption>Normal Projection</caption -->
  <adjust>right</adjust>
  <up_path>object_images/p001.png</up_path>
  <over_path>object_images/p002.png</over_path>
  <down_path>object_images/p003.png</down_path>
  <action>projection0</action>
</object>
<object>
  <pos_x>25</pos_x>
  <pos_y>150</pos_y>
  <!-- caption>Semi Stereographic Projection</caption -->
  <adjust>right</adjust>
  <up_path>object_images/p101.png</up_path>
  <over_path>object_images/p102.png</over_path>
  <down_path>object_images/p103.png</down_path>
  <action>projection1</action>
</object>
<object>
  <pos_x>50</pos_x>
  <pos_y>150</pos_y>
  <!-- caption>Stereographic Projection</caption -->
  <adjust>right</adjust>
  <up_path>object_images/p201.png</up_path>
  <over_path>object_images/p202.png</over_path>
  <down_path>object_images/p203.png</down_path>
  <action>projection2</action>
</object>
<object>
  <pos_x>25</pos_x>
  <pos_y>175</pos_y>
  <!-- caption>View Experience</caption -->
  <adjust>right</adjust>
  <up_path>object_images/ve01.png</up_path>
  <over_path>object_images/ve02.png</over_path>
  <down_path>object_images/ve03.png</down_path>
  <action>view_experience</action>
</object>
</panorama_controller>
```

End of document

This document is for informational purposes only. Ryubin Office makes no warranties, express, implied, or statutory as to the information in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, translated, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Ryubin Office.

Ryubin Office may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Ryubin Office, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2011 Ryubin Office.

All rights reserved. Your right to copy or translate this documentation is limited by copyright law. Making unauthorized copies, adaptations, compilations, translations or derivative works for commercial distribution is prohibited and constitutes a violation of the law.