ThumbnailSlider Component

Technical Documentation

ThumbnailSlider is a navigation component that enables you to load a set of images,swf's or symbols from library,fed from an xml file ,through which you can scroll with the help of the arrows,or simply depending on the mouse position.

It's an excellent component when used in a photo gallery for example, because it was developed to adjust on any environment you could think,technical and artistic, due to the possibility of setting the slider's orientation to vertical positioning,or horizontal,due to the roll over effects on the thumbs that will add creativity and style to your project,due to the resizing types supported by the thumbs that will enable you to display the content in the best way possible.

ThumbnailSlider can be accessed from Components panel under UI Pro Components – jumpEYE.

Properties

arrowsControl(**enumeration="click,rollOver,mouseMove",defaultValue="click"**)-determines the type of control over the movement of the loaded thumbs.
baseUrl(**default=""**)-optional property that you can use it in case you want to complete the path to the thumbnails in the xml file,and let the attribute "thumbnail" only with the name of the file;
border(**default=true**)-property type Boolean; if true,the thumbs will have a border;false will load the content without a border.
borderColor(**default=#FFFFFF,verbose=1**)-case the border parameter is set true,you can set the border's color;this property will be visible only in Component Inspector panel.
borderCornerRadius(**default=0,verbose=1**)-property that sets the border's corner radius;this property will be visible only in Component Inspector panel.
borderSize(**default=5**)-case border parameter is set true,you can set the border's size;the maximum size is 10 and the minimum size=0.
builtInPreloader(**enumeration=none,bar,circular,default=none**)-choose the built in preloader between a bar preloader,circular preloader or no preloader.

easeType(**enumeration="Back,Bounce,Elastic,Regular,Strong,None",defaultValue="Strong",verbose=1**)property type String that will set the easeType in the slider's movement;this property will be visible only in Component Inspector panel.
effectAmount(**default=50,verbose=1**)-property that will set the amount of effect on the thumbs when the rollOver event will be triggered;this property will be visible only in Component Inspector panel.
effectTimeIn(**default=10,verbose=1**)-property that will set the duration effect on the thumbs when the rollOver event will be triggered;this property will be visible only in Component Inspector panel.
effectTimeOut(**default=10,verbose=1**)-property that will set the duration effect on the thumbs when the rollOut event will be triggered;this property will be visible only in Component Inspector panel.
inactiveCenterArea(**default=20,verbose=1**)-property that is used in the case of arrowsControl=mouseMove and defines a percentage from the total width of the slider where when the mouse is moving the slider will not move;
rollOverEffect(**enumeration="none,blur,grayscale,alpha,brightness,distortion,colorLight,scaleBorder,scale",defaultValue="none"**)-choose the type of roll over effect on the thumbs.
orientation(**enumeration="horizontal,vertical",defaultValue="horizontal"**)choose how the thumbs in the slider will be displayed
preloaderColor(**default=#336699,verbose=1**)-case the builtInPreloader parameter is set to bar or circular,you can set the preloader's color;this property will be visible only in Component Inspector panel
remainSelected(default=true,verbose=1)-case the remainSelected property is set to true the selected thumb will remain selected;this property will be visible only in Component Inspector panel
resizeType(**enumeration=resize,scale,crop,noscale,default=resize**)-set's the thumbs dimensions according to the type of resize you choose.
reverseRollOverEffect(**default=false,verbose=1**)property type boolean that enables the reverse of the rollOverEffect;this property will be visible only in Component Inspector panel.
spacing(**defaultValue=5**)-property type number that will set the distance between the thumbs.
targetDisplay(**defaultValue=""**)-optional property type String that you can use if you want to automatically load the images from the into a loader that has the property contentPath onReleaseThumb
thumbHeight(**defaultValue=100**)-property type number that will set the thumbs height;if the orientation will be set to horizontal,the thumbs height will be determined by the slider's height.
thumbWidth(**defaultValue=120**)-property type number that will set the thumbs width;if the orientation will be set to vertical,the thumbs width will be determined by the slider's width.
tweeningSpeed(**defaultValue=10**)-property type number that will set the slider's movement speed.
xmlPath(**defaultValue=""**)-the path to the xml that will be loaded.

Methods

load(**arg:String**)-you can use this method if you want to change the xml path to load another content in the slider.
setSize(**width:Number**,**height:Number**)-you can use this method to change the slider's dimensions;if the slider's orientation is set to horizontal, then thumbs height will be determined by the slider's height; if the slider's orientation is set to vertical, then thumbs

width will be determined by the slider's width.
startSlideshow(**time:Number**)-you can start an auto thumb selection that is very helpful when trying to make a photo gallery.It's recommended to start the slide show only when the content has been loaded,that is when the onLoadComplete event returns a true value.Use this method in combination with the event onSlideshowChange which will return the selected object;
pauseSlideshow()-method used to make a hold in the auto thumb selection.
resumeSlideshow()-method used to resume the slide show after it was paused.
stopSlideshow()-method used to stop the slide show.
selectNextThumb()-method that enables the selection of the next thumb.
selectPreviousThumb()-method that enables the selection of the previous thumb.
stopSlideshow() - stops the slideshow

Events

onLoadXml(arg:Boolean)-returns a boolean value which is true if the xml has loaded and false if an error occurred and the xml couldn't load.
onLoadProgress(loadedFiles,totalFiles)-returns the number of loaded external content in the slider and the total number of external contents in the slider.
onLoadComplete(arg:Boolean)-returns a boolean value which is true if the contents has been loaded and false if an error has occur during the loading process or the contents are still loading
onPressThumb(arg:Object,container:String,index:Number)-returns an object with the properties as the attributes of the xml nodes when a thumb is pressed ,container which is the reference of the content's container,and index as the thumb's index in the slider
onReleaseThumb(arg:Object,container:String,index:Number)-returns an object with the properties as the attributes of the xml nodes when a thumb is released,container which is the reference of the content's container,and index as the thumb's index in the slider
onDragOutThumb(arg:Object,container:String,index:Number)-returns an object with the properties as the attributes of the xml nodes when a thumb is dragged out ,container which is the reference of the content's container,and index as the thumb's index in the slider
onReleaseOutsideThumb(arg:Object,container:String,index:Number)-returns an object with the properties as the attributes of the xml nodes when a thumb is released outside ,container which is the reference of the content's container,and index as the thumb's index in the slider
onRollOverThumb(arg:Object,container:String,index:Number)-returns an object with the properties as the attributes of the xml nodes when a thumb is rolled over ,container which is the reference of the content's container,and index as the thumb's index in the slider
onRollOutThumb(arg:Object,container:String,index:Number)-returns an object with the properties as the attributes of the xml nodes when a thumb is rolled out ,container which is the reference of the content's container,and index as the thumb's index in the slider
onSlideshowChange(arg:Object,container:String,index:Number)returns an object with the properties as the attributes of the xml nodes when a thumb is selected from the slide show ,container which is the reference of the content's container,and index as the thumb's index in the slider

Notes

Case the parameter arrowsControl is set to “click” or “rollOver” ,you can change the arrows by making your own symbols with the linkage id "sliderBackHorizontal" and

"sliderForwardHorizontal" if the orientation is set to horizontal,and "sliderBackVertical" and "sliderForwardVertical" if orientation is set to vertical.

The frame rate should be kept above the 20 frames/second minimum limit,best would be 35.

The attribute thumbnail from the xml file is a necessary parameter which is used for the component to know what to load in the thumbs;this attribute can be the full path to the file or just the name of the file and the rest of the path completed with the baseUrl property;also is broadcasted as a property of an object in the components events like onPressThumb, onRollOverThumb,onRollOutThumb,onReleaseThumb,onReleaseOutsideThumb,onDragOutThumb,onSlideshowChange.

The attribute large from the xml file is an optional parameter case you want to load a large image that coresponds to the selected thumbnail into a specific container ,parameter that is broadcasted as a property of an object in the components events like onPressThumb,onRollOverThumb,onRollOutThumb,onReleaseThumb,onReleaseOutsideThumb,onDragOutThumb,onSlideshowChange.

The attribute thumbEvents from the xml file is used for the component to know if the events on the thumbs will be active;case false,the onRollOverEffects parameter will be disabled;

The attribute description from the xml file is an optional parameter which is broadcasted as a property of an object in the components events like onPressThumb, onRollOverThumb,onRollOutThumb,onReleaseThumb,onReleaseOutsideThumb,onDragOutThumb,onSlideshowChange.

You can place an attribute url if you want to open an url in a blank window when you make release on thumb;

XML

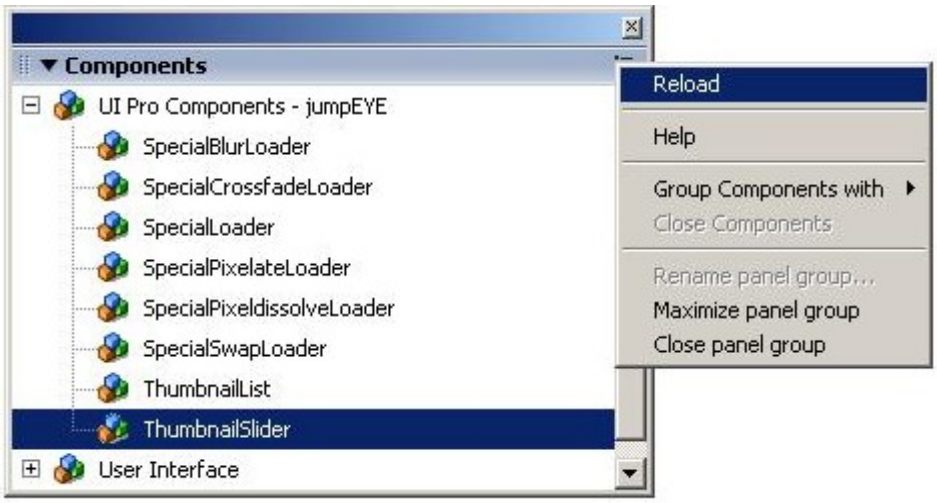
This is how the xml file that is loaded by the component should look:

```
<?xml  version='1.0'  encoding='UTF-8'?>
<gallery>
    <img thumbnail='pictures/thumbnails/01.jpg' large='pictures/large/01.jpg'
description='image description' url='http://www.flashstore.com'/>
    <img thumbnail='pictures/thumbnails/02.jpg' large='pictures/large/02.jpg'
description='image description' url='http://www.flashstore.com'/>
    <img thumbnail='pictures/thumbnails/03.jpg' large='pictures/large/03.jpg'
description='image description'/>
    <img thumbnail='pictures/thumbnails/04.jpg' large='pictures/large/04.jpg'
description='image description'/>
    <img thumbnail='pictures/thumbnails/05.jpg' large='pictures/large/05.jpg'
description='image description'/>
    <img thumbnail='pictures/thumbnails/06.jpg' large='pictures/large/06.jpg'
description='image description'/>
    <img thumbnail='pictures/thumbnails/07.jpg' large='pictures/large/07.jpg'
description='image description'/>
    <img thumbnail='pictures/thumbnails/08.jpg' large='pictures/large/08.jpg'
description='image description'/>
    <img thumbnail='pictures/thumbnails/09.jpg' large='pictures/large/09.jpg'
description='image description'/>
    <img thumbnail='pictures/thumbnails/10.jpg' large='pictures/large/10.jpg'
description='image description'/>
    <img thumbnail='pictures/thumbnails/11.jpg' large='pictures/large/11.jpg'
description='image description'/>
    <img thumbnail='pictures/thumbnails/20.jpg' large='pictures/large/20.jpg'
description='image description'/>
     <img  thumbnail='pictures/thumbnails/21.jpg'  large='pictures/large/21.jpg'
description='image description'/>
</gallery>
```

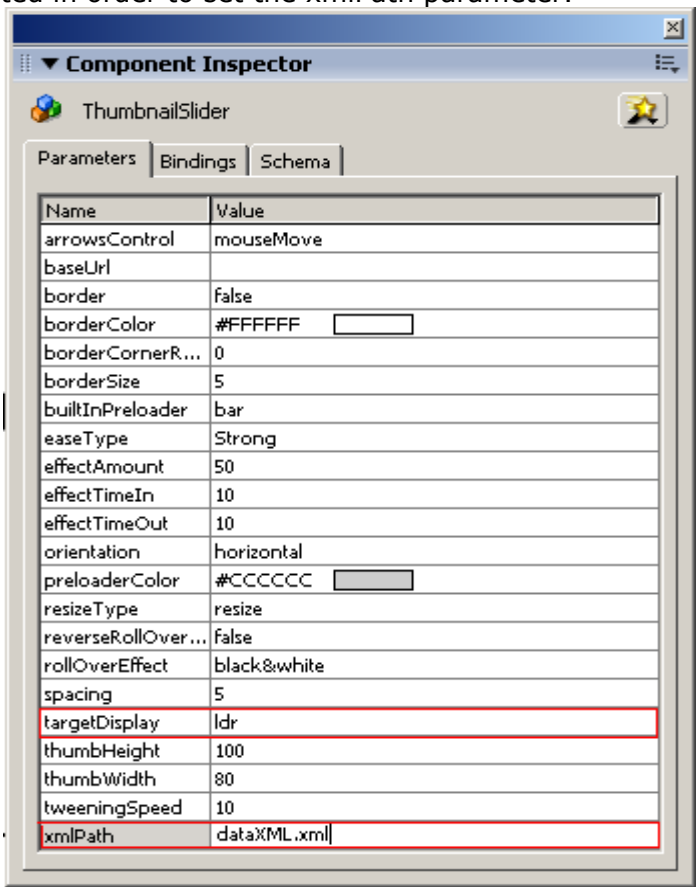
You can write this in any text editor and then save it under the extension xml,like XMLData.xml for example.

Tutorial 1-Simple "Put in stage and compile"

In this tutorial we are going to build a simple photo gallery that uses the ThumbnailSlider component and a loader component in our case the SpecialCrossfadeLoader.
First you have to instal the mxp file that you received and then close Flash and open it again to refresh the Components panel or just press Ctrl+F7 and press the reload button.



So,with the ThumbnailSlider component in your Components list drag it onto the Stage.First of all you should have the Component Inspectorr(Alt+F7) or the Properties panel opened and the ThumbnailSlider selected in order to set the xmlPath parameter.



Next we will drag the SpecialCrossfadeLoader onto the Stage and just give it an instance name "ldr" and then we set the ThumbnailSlider's property targetDisplay to ldr.This is how we make the connection between the ThumbnailSlider and the SpecialCrossfadeLoader so that when you press a thumbnail a picture corresponding with that thumbnail is displayed in our loader.

Now all we have to do is press Ctrl+Enter to test our movie.

Tutorial 2-Start a slide show-auto scroll

In this tutorial we are going to use a function provided by the component in order to start an slide show. Once you have the component in stage with an instanceNmae all you have to do is write the code bellow:

```
var listn:Object = new Object();
listn.onLoadComplete = function(loaded) {
	if (loaded) {
		sld.startSlideshow(3);
	}
};
sld.addListener(listn)
```

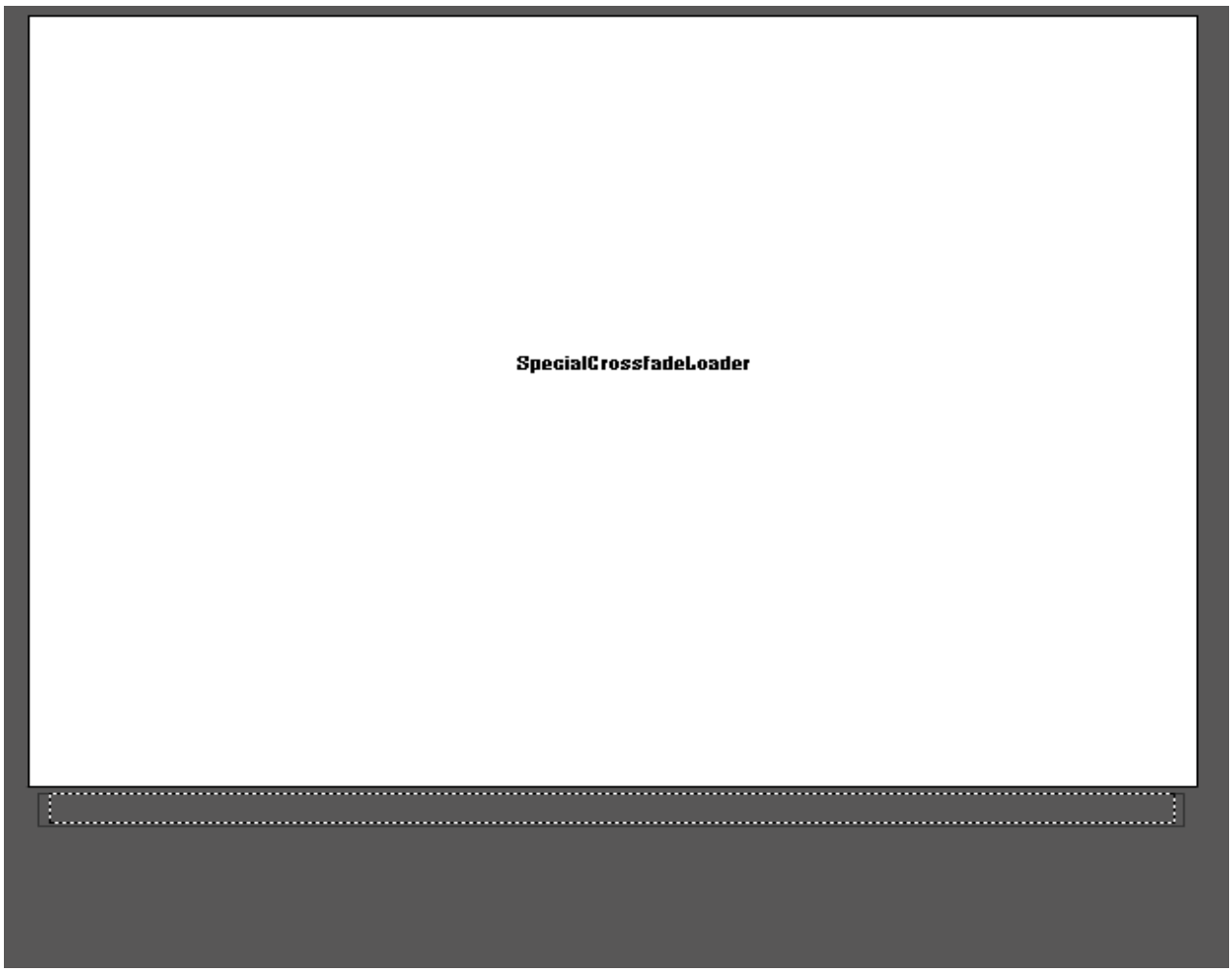
In this tutorial we gave the component on stage the instanceName "sld".When all the content is loaded the slideshow starts and auto scrolls throw the thumbnails at a time interval of 3 seconds.

You can use this code sample in combination with the onSlideshowChange if you want something to happened during the slideshow,maybe load the selected thumb in to a loader.And the code sample for this would look like this:

```
listn.onSlideshowChange = function(obj, arg, ind) {
	loader.contentPath = obj.thumbnail;
};
sld.addListener(listn)
```

Tutorial 2-Attaching the component from script

In this tutorial we will make a link between the ThumbnailSlider component and the SpecialCrossfadeLoader(http://store.jumpeye.com/crossfade_loader/index.html) to make a photo gallery.First off all make sure that you have installed the mxp extension for the ThumbnailSlider.If you don't have the SpecialCrossfadeLoader you can use another loader instead and then set the Stage at 650x550 px and the movie frames/second to 35.We will also need a text box with the instance name description that will place under the loader and which will display a short description for every picture ,description that is set in the xml file at the attribute description(see the XML format above).The images for the ThumbnailSlider will be put in to a directory named "thumbnails" and the ones for the big loader in a directory named "large" .You can place your pictures anywhere as long as the filepath that you write in the xml file is correct.



Now you need to have the loader component on Stage or attach it if you want,and the ThumbnailSlider component in the library.We are going to use the attachMovie function in order to have the an instance of our component on Stage ,and give the loader component already on the Stage an instance name like "ldr".We are using the atachMovie fuction in the case of the ThumbnailSlider component because we want to show you how you can do this.
So just type in your action script window the following code:

```
_root.attachMovie("ThumbnailSlider", "slider", _root.getNextHighestDepth(),{_x:18,_y:480});
with (_root.slider)
{
	arrowsControl = "click";
	border = true;
	borderSize = 5;
	baseUrl = "";
	borderCornerRadius = 5;
	tweeningSpeed = 10;
	easeType = "Strong";
	builtInPreloader = "bar";
	preloaderColor = "0x000000";
	borderColor = "0xFFFFFF";
	orientation = "horizontal";
	effectTimeIn = 10;
	effectTimeOut = 10;
	effectAmount = 50;
	rollOverEffect = "colorLight";
```

```
        reverseRollOverEffect = false;
        spacing = 6;
        xmlPath = "dataXML.xml";
        resizeType = "resize";
        thumbWidth = 80;
        thumbHeight = 60;
}
_root.slider.setSize(600,60);
```

This will attach the ThumbnailSlider on the _root level.Next we are going to listen the component to see when the user is choosing a thumb that will be displayed in our SpecialCrossfadeLoader and we are going to start a slideshow after the pictures are all loaded For this we will work with the events that the ThumbnailSlider provide(look at Events):

```
var listEvents:Object=new Object();

listEvents.onLoadComplete = function(loaded) {
        if (loaded) {
                _root.slider.startSlideshow(5)
        }
};

listEvents.onReleaseThumb = function(obj:Object, arg, index) {
        _root.ldr.contentPath = obj.large;
        _root.description.text = obj.description;
};
slider.addListener(listEvents)
```

All you have to do now is press Ctrl+Enter to test your movie;