MONO
SLIDESHOW

Monoslideshow Manual
version 3.1
revision 1

Contents

Introduction

Monoslideshow is an ultra customizable HTML5 / Javascript image and video viewer. It can be embedded in every website, in whatever size you want. It's designed for maximum flexibility, whether you want to have a professional looking portfolio with cinematic effects, or a basic image slideshow.

Key features

- Professional transitions
- Ken Burns effect
- Smart group layouting
- Customizable interface elements
- Built-in photo browser
- Fully responsive and scalable
- Compatible with the latest popular browsers
- Compatible with iPhone and iPad
- Javascript API
- Completely stand-alone, no other Javascript libraries needed

What you'll need

- Basic HTML / Javascript knowledge
- A text editor (e.g. Notepad, Sublime Text)
- A website to embed the slideshow in

How does it work?

The core of Monoslideshow is a single Javascript file. There are no external files like interface artwork or CSS stylesheets. All Monoslideshow needs is a folder with images and a single .XML-file.

The .XML-file is formatted according to the specifications in this document. It contains all the content and layout information. This setup makes Monoslideshow compact, extremely versatile and easy for web developers to integrate in existing websites.

The basics

A single .XML file to customize

Monoslideshow is a single Javascript file that uses information from a XML file to configure the contents and layout of the slideshow. XML files are not difficult to master, think of the XML language as HTML with specific tag names. If you know basic HTML code, then you know how XML works.

Tip: use the demo page on www.monoslideshow.com/demo to automatically set up and download XML files as a starting point for further development.

A brief introduction to XML

Monoslideshow reads all its information from a single file called `monoslideshow.xml`. To put images in your slideshow, you have to edit this file. This can be done by using any text editor.

XML is a computer language, and actually it’s a bit like HTML. Just as with HTML, content in XML files is placed inside tags.

Example 1

Let’s take a look at an example to illustrate how all this works.

```
<?xml version="1.0" encoding="utf-8"?>
<album thumbnail="albumThumbnail.jpg" itemPath="images">
    <contents>
        <image source="photo1.jpg" thumbnail="thumb1.jpg" />
        <image source="photo2.jpg" thumbnail="thumb2.jpg" />
        <image source="photo3.jpg" thumbnail="thumb3.jpg" />
    </contents>
</album>
```

The first line says that this textfile is a file that uses the XML-language. You always have to include this line. The second line starts with an opening tag `<album>`. It’s corresponding closing tag is located at the last line. Everything between these two tags are its contents. An opening tag must always have a matching closing tag. The only way a closing tag differs from its opening tag, is an extra forward slash.

Inside the opening tag `<album>` are two attributes, called `thumbnail` and `itemPath`. Attributes are always placed inside a tag. The value of an attribute (in this case `albumThumbnail.jpg` and `images`, respectively) must always be enclosed between quotes ("). They are used to specify something about the tag they are contained in. In this case, the thumbnail of the album is `albumThumbnail.jpg` and the path that contains the images is `images`.

The three next `<image>` tags are the actual image files. They all contain two attributes, `source` and `thumbnail`. The attribute `source` specifies the link to the image inside the folder specified by `itemPath`. The link to the thumbnail is specified by the attribute `thumbnail`. As you’ve perhaps noticed, the `<image>` tags lack a regular closing tag. Instead of writing `<image source="photo1.jpg"></image>`, you can also write `<image source="photo1.jpg" />`. This type of tag is called a "self closing tag". We use this type here because the `<image>` tag has no content by itself. All the information is contained in its attributes.

You can completely build your own slideshow by writing your own .XML file. The options are endless, and are all discussed in the XML reference section of this document.

Example 2

Let’s take a look at another example:

```
<?xml version="1.0" encoding="utf-8"?>
<album>
    <configuration>
        <transition type="push" delay="3" />
    </configuration>
    <contents>
        <album thumbnail="albumThumbnail1.jpg" itemPath="album1">
            <contents>
                <image source="photo1.jpg" thumbnail="thumb1.jpg" />
                <image source="photo2.jpg" thumbnail="thumb2.jpg" />
                <image source="photo3.jpg" thumbnail="thumb3.jpg" />
            </contents>
        </album>
        <album thumbnail="albumThumbnail2.jpg" itemPath="album2" thumbnailPath="albumThumbnails2" >
            <contents>
                <image source="photo1.jpg" />
                <image source="photo2.jpg" />
                <image source="photo3.jpg" />
                <image source="photo4.jpg" />
                <image source="photo5.jpg" />
            </contents>
        </album>
    </contents>
</album>
```

This slideshow consists of two albums. You can place `<album>` tags inside other album's `<contents>` tags to create sub albums. In this way, you can also create sub-sub albums, and you can go to even deeper levels as well.

Two new tags are introduced here: the `<configuration>` and the `<transition>` tag. Here you'll insert the configuration attributes that apply for the elements inside the `<contents>` tag of this album. In this case, the delay is set to three seconds and the transition is set to type "push".

Attributes are recursive

Attributes also apply to sub albums. However, you can define a `<configuration>` tag in each `<album>` tag, so that each album can have its own configuration. Configuration attributes inserted at a lower level override those at a higher level. In this case, there's only one configuration defined, and it applies to both its sub albums.

Smart file references

In example 2, a new attribute is introduced in the second `<album>` tag, called `thumbnailPath`. This attribute defines the path to all the thumbnail images. Now, if an `<image>` tag doesn’t contain a `thumbnail` attribute, it looks for a thumbnail file in the folder `albumThumbnails2` with the exact same name as defined in the `source` attribute of that image. This way, you can easily set up two folders: One for all the images (in this case `album2`), and one for all the thumbnails (in this case `albumThumbnails2`). Each folder then contains the exact same file names, but one is used for thumbnails and the other for the full size images.

Implementation

Monoslideshow is made to be implemented in a web page. You can do this by inserting code in a HTML page. As a global overview, there's three steps you have to follow:

- preparing images and folder structure
- creating .XML file
- putting Monoslideshow on your site

Preparing images and folder structure

Monoslideshow uses your full resolution images to display in the slideshow viewer and uses low resolution thumbnails to display in the image navigation browser. You have to prepare these files yourself. Creating thumbnails one by one is a tedious process that is easily avoided by applying batch processes. Batch processes take multiple files and apply the same set of actions to them. For example, resizing images down to a lower resolution. Please refer to your image editing application on how to do this.

Once the files are in place, this could be a typical folder structure:

```
/index.html
/monoslideshow.xml
/slideshow/monoslideshow.js
/slideshow/images/
/slideshow/images/photo1.jpg
/slideshow/images/photo2.jpg
/slideshow/images/photo3.jpg
/slideshow/thumbnails/
/slideshow/thumbnails/photo1.jpg
/slideshow/thumbnails/photo2.jpg
/slideshow/thumbnails/photo3.jpg
```

Creating your .XML file

Once all image files are in place, you can put the references to all the images and thumbnails in your .XML file. You can then extend your .XML file by inserting titles, descriptions and by modifying layout settings. There's a multitude of options available for you to adjust. Please use the examples and the XML reference in this document to see how the .XML file is structured and which options are available. You can also generate a .XML file as a starting point for your own on the demo page: monoslideshow.com/demo.

Putting Monoslideshow on your site

To put Monoslideshow on your site, you have to embed it in your web page. Below is an example of how to do this:

```
<html>
    <head>
        <title>Monoslideshow</title>
        <script type="text/javascript" src="slideshow/monoslideshow.js"></script>
    </head>
    <body onload="onLoadComplete();">
        <div id="monoslideshowHolder" style="width: 640px; height: 480px;"></div>
        <script type="text/javascript">
            function onLoadComplete() {
                var mss = new Monoslideshow('monoslideshowHolder');
                mss.load('slideshow/monoslideshow.xml');
            }
        </script>
    </body>
</html>
```

The Monoslideshow Javascript file `monoslideshow.js` is placed in the `<head>` tag of your website's HTML code. When your website is loaded, the function `onLoadComplete` is called (as set by the `onLoad` attribute of the `body` tag). This function instantiates Monoslideshow as variable `mss` and tells it to load itself in the `div` tag with `id` attribute `monoslideshowHolder`. The second line in this function finally tells Monoslideshow to load the .XML file `slideshow/monoslideshow.xml`.

Customization

Monoslideshow can be customized by altering the .XML file, and by communicating with the Monoslideshow Javascript API. In this chapter, a couple of the most typical customizations are presented.

Smart group layouting

Monoslideshow has support for smart group layouting. You can specify multiple images in a `group` tag and the slideshow will automatically find the optimal layout for the images it contains. The following slideshow contains two slideshow items, a group with two and another group with three images:

```
<?xml version="1.0" encoding="utf-8"?>
<album thumbnail="albumThumbnail.jpg" itemPath="images">
    <contents>
        <group>
            <contents>
                <image source="photo1.jpg" thumbnail="thumb1.jpg" />
                <image source="photo2.jpg" thumbnail="thumb2.jpg" />
            </contents>
        </group>
        <group>
            <contents>
                <image source="photo3.jpg" thumbnail="thumb3.jpg" />
                <image source="photo4.jpg" thumbnail="thumb4.jpg" />
                <image source="photo5.jpg" thumbnail="thumb5.jpg" />
            </contents>
        </group>
    </contents>
</album>
```

The `contents` tag of a `group` tag can only contain `image` tags. `video` tags or other `group` tags are not supported.

Making Monoslideshow responsive

Monoslideshow is fully resizable. This makes Monoslideshow an ideal slideshow component for your responsive layouts. Whether Monoslideshow is viewed on your laptop, tablet or phone, it always displays its content in the most optimal way. The only thing that has to be done is telling Monoslideshow to resize itself to the dimensions you want. The example below updates the dimensions of Monoslideshow every time the browser window is resized:

```
<html>
    <head>
        <title>Monoslideshow</title>
        <script type="text/javascript" src="slideshow/monoslideshow.js"></script>
    </head>
    <body onload="onLoadComplete();">
        <div id="monoslideshowHolder"></div>
        <script type="text/javascript">
            function onLoadComplete() {
                var mss = new Monoslideshow('monoslideshowHolder');
                function resizeMonoslideshow () {
                    mss.resize(window.innerWidth, window.innerHeight);
                }
                window.onresize = resizeMonoslideshow;
                resizeMonoslideshow();
                mss.load('slideshow/monoslideshow.xml');
            }
        </script>
    </body>
</html>
```

Monoslideshow even has support for different configurations per dimension specification, much like CSS Media Queries. For example, have a look at this .XML file configuration:

```
<?xml version="1.0" encoding="utf-8"?>
<album>
    <configurations>
        <variant maxWidth="480">
            <transition type="cube"/>
        </variant>
        <variant minWidth="481" maxWidth="1024">
            <transition type="push"/>
        </variant>
    </configuration>
    <contents>
        <image source="photo1.jpg" />
        <image source="photo2.jpg" />
        <image source="photo3.jpg" />
    </contents>
</album>
```

Instead of one `configuration` tag, there are now two `variant` tags inside a `configurations` tag. These `variant` tags can contain all the tags and attributes that normal `configuration` tags can contain. There's one important distinction: they override normal configuration settings, because they are targeted more specifcally. That is because of their size selection criteria. The criteria attributes applied in this case are `minWidth` and `maxWidth`. The above settings state: when Monoslideshow is 480 pixels wide at a maximum, apply the transition type `cube`, in between 481 and 1024 pixels wide, apply transition type `push`, and otherwise apply the default transition (which is `blend`).

Making different configuration variants based on size selection criteria is a great way to ensure that your slideshow works perfectly on all devices you are targeting.

Using multiple sources for one image

When making Monoslideshow responsive, it's often desired to have a different set of images for a different type of screens. For example, it's no use loading a high resolution image on a mobile device, as it's just wasting bandwidth. You can target devices with multiple image sources in a couple of ways. Let's take a look at image variants first:

```
<album>
    <contents>
        <image>
            <sources>
                <variant source="photo1a.jpg" width="320" />
                <variant source="photo1b.jpg" width="640" />
            </sources>
        </image>
        <image>
            <sources>
                <variant source="photo2a.jpg" width="320" />
                <variant source="photo2b.jpg" width="640" />
            </sources>
        </image>
    </contents>
</album>
```

This example shows two image tags, each having two different file sources. The `width` attribute of the image tags specifies in which case that specific source variant has to be displayed. When the slideshow is only 300 pixels wide, the first variant (with width 320) is chosen. If the slideshow is wider than 320 pixels, the second variant (with width 640) is chosen.

Similarly, you can use the `height` attribute to target the image variants to multiple devices, or you can use both. Note that the `width` and `height` attributes do not have to be the exact same dimensions of the images themselves. They merely are an instruction for Monoslideshow for when to use which image variant.

Specifying source variants is a great way to make Monoslideshow fully responsive to a wide array of devices. having multiple variants of the same image has the added benefit that Monoslideshow only loads the most optimal image for the current slide. For example, if smart group layouting is selected, there are multiple images on one slide, and Monoslideshow automatically chooses the best fitting image, i.e. the image with the least superfluous pixels for the frame it's displayed in, but never less that the frame's dimensions.

Another, more concise way to have multiple source files for one image, is to make use of the `filenameSuffix` attribute. This attribute appends text after the file's basename and before the file extension. Combining it with configuration variants, you'll end up with the configuration below. On smaller screens, the source "photo1.jpg" will be loaded, and for larger screens the source "photo1@2x.jpg".

```
<album>
    <configurations>
        <variant maxWidth="320" />
        <variant minWidth="321" filenameSuffix="@2x" />
    </configuration>
    <contents>
        <image source="photo1.jpg" />
        <image source="photo2.jpg" />
        <image source="photo3.jpg" />
    </contents>
</album>
```

If you prefer to have your image variants across different folders, then you can use a setup like this:

```
<album>
    <configurations>
        <variant maxWidth="320" itemPath="lowres" />
        <variant minWidth="321" itemPath="highres" />
    </configuration>
    <contents>
        <image source="photo1.jpg" />
        <image source="photo2.jpg" />
        <image source="photo3.jpg" />
    </contents>
</album>
```

Starting with a specified item or album

You can start your slideshow with a specified item or album by instructing Monoslideshow to look for a .XML node with a specific `id`.attribute. You can do this by specifying `startWithImageID` or `startWithAlbumID` in the album's `configuration` node. For example, the following code instructs Monoslideshow to start with the item `myPhoto`:

```
<album>
    <configuration startWithItemID="myPhoto" />
    <contents>
        <image source="photo1.jpg" />
        <image source="photo2.jpg" id="myPhoto" />
        <image source="photo3.jpg" />
    </contents>
</album>
```

The next example shows how to start with the album `myAlbum`:

```
<album>
    <configuration startWithAlbumID="myAlbum" />
    <contents>
        <album>
            <contents>
                <image source="photo1.jpg" />
            </contents>
        </album>
        <album id="myAlbum">
            <contents>
                <image source="photo2.jpg" />
            </contents>
        </album>
    </contents>
</album>
```

It does not matter how deeply nested the item with the specified `id` is, as long as you define `startWithItemID` or `startWithAlbumID` in the top album configuration node.

Customizing captions with EXIF data

By default, Monoslideshow displays a caption element over the image. It uses the `title` and `description` attributes of the `image` tags for that. Monoslideshow also provides a caption template system for defining which meta information is displayed in the title and description.

The caption template is defined by the `template` attribute in the `title` and `description` tag. This attribute is formatted as text that holds certain variables. These variables are enclosed within curly brackets. Here is an example of templates for titles and descriptions:

```
<caption>
    <title template="{index}/{albumSize}: {title} - (shot with aperture f/{FNumber})" />
    <description template="[{Copyright}, ]shot using: {Make} {Model}" />
</caption>
```

As you can see, there a few variables defined between curly brackets. In the example above, the title template defines that titles always start with the current index, followed by the total album size. Then the title as defined in the image tag is displayed, with information about the aperture value in parentheses. The description template defines that all descriptions start with the copyright notice of the current image, followed by information about the equipment used to shot it. It doesn't insert the original description.

You can use square brackets to define groups. If a variable inside a group doesn't contain any information, the entire group is left out.

Monoslideshow fills in the defined variables with built in information and EXIF data (if present in the image file). Have a look at the XML reference and the EXIF reference to see which variables you can use.

Disabling start up logo

By default, Monoslideshow shows a logo during the initialization process of the slideshow. During this process the .XML file is loaded and the layout is initialized. To disable the Monoslideshow logo, you have to pass the variable `showLogo` to Monoslideshow and set it to "false":

```
var mss = new Monoslideshow('monoslideshowHolder', {showLogo: false});
mss.load('slideshow/monoslideshow.xml');
```

Disabling registration info

By default, Monoslideshow shows registration info in the right-click context menu. To disable this, you have to pass the variable `showRegistration` to Monoslideshow and set it to "false":

```
var mss = new Monoslideshow('monoslideshowHolder', {showRegistration: false});
mss.load('slideshow/monoslideshow.xml');
```

Setting a base folder

Sometimes it's convenient to set a base folder to which all files referenced in the .XML file are relative to. If the base attribute is set, all files in the .XML file are prepended by the folder defined in the base tag. Be sure to end the base attribute with a folder separator (forward slash).

```
var mss = new Monoslideshow('monoslideshowHolder', {base: 'myCustomFolder/'});
mss.load('slideshow/monoslideshow.xml');
```

Monoslideshow API

Monoslideshow has a Javascript API, so you can control Monoslideshow in your own scripts. Monoslideshow can receive function calls that instruct the slideshow to perform certain actions. Monoslideshow also broadcasts events, that your program can listen to. This way, you can synchronize your actions to Monoslideshow events, and Monoslideshow can perform actions on your command.

API calls

The following methods are available to any instantiated Monoslideshow object:

method / attribute	description
`addEventListener(eventName, function)`	adds a listener function to event
`enterFullScreen()`	enters full screen
`exitFullScreen()`	exits full screen
`goTo(index)`	jumps to item index
`isPlaying()`	returns `true` if the slideshow is playing, `false` otherwise
`load(file)`	loads .XML file
`loadXMLString(text)`	loads a .XML string
`navigationClose()`	closes the navigation window
`navigationNext()`	navigates to next page in navigation window
`navigationOpen()`	opens the navigation window
`navigationPrevious()`	navigates to previous page in navigation window
`navigationUp()`	navigates to parent album
`next()`	navigates to next item in the slideshow
`pause()`	pauses the slideshow
`play()`	start playing the slideshow
`previous()`	navigates to previous item in the slideshow
`removeEventListener(eventName, function)`	removes listener function for event
`reset()`	restarts the slideshow and reloads the XML.
`resize(width, height)`	resizes the slideshow to the dimensions `width` and `height`
`version`	text containing current version

Events

The following events are available to listen to:

event	attributes	description
`albumEnd`	`default`	the current album finished displaying its last slideshow item
`error`		an error occurred
`exitFullScreenSelect`		select exit full screen
`fullScreenSelect`		select full screen
`itemLoadComplete`	`default`	item has completed loading
`itemLoadError`	`default`	an error occurred during item load
`itemLoadStart`	`default`	item has starting loading
`itemPreloadComplete`	`default`	item has completed preloading
`itemPreloadError`	`default`	an error occurred during item preload
`itemPreloadStart`	`default`	item has starting preloading
`monoslideshowInitialized`		Monoslideshow has initialized itself and is ready to receive API calls
`navigationClose`		navigation has closed
`navigationCloseSelect`		initiate navigation close
`navigationItemSelect`	`default`	select navigation item

navigationNextSelect		next page in navigation window
navigationOpen		navigation has opened
navigationOpenSelect		initiate navigation open
navigationPreviousSelect		previous page in navigation window
nextSelect		select next item
pauseSelect		select slideshow pause
playSelect		select slideshow play
previousSelect		select previous item
thumbnailNavigationItemSelect	default	select thumbnail navigation item
transitionEnd	default	transition sequence has ended
transitionOutroStart	default	transition sequence is about to start outro phase (when performOutro is set)
transitionStart	default	transition sequence is about to begin

Default event attributes

The following attributes are available by default when Monoslideshow broadcasts an event:

attribute	type	description

album	object {
	description: text,
	id: text,
	size: number,
	title: text
	}
		object containing current album variables

data	text	user-defined data string, useful for passing custom data from the .XML file to Javascript listeners
definedSources	array containing strings	sources as defined in the .XML file
description	text	description of current item

group	array containing objects {
	definedSources: array,
	sources: array
	}
		object containing current album variables

id	text	id of current item
index	number	position in the current album
link	text	link to navigate to when clicked on the item
linkTarget	text	target to load the link into

sources	array containing objects {
	url: text
	type: text
	}
		sources as parsed by Monoslideshow. When item is a video, type contains video type.

title	text	title of current item
type	text	type of current item: video, image or group. When type is group, the attributes definedSources and sources are contained in an object called group.

Examples

Passing XML data as a Javascript string

You can directly pass a Monoslideshow formatted XML string to the Monoslideshow instance. Example:

```
var mss = new Monoslideshow('monoslideshowHolder');
mss.loadXMLString(
    '<album>' +
        '<contents>' +
            '<image source="1.jpg" />' +
            '<image source="2.jpg" />' +
            '<image source="3.jpg" />' +
        '</contents>' +
    '</album>'
);
```

Listening to events

To listen to Monoslideshow events in Javascript, you have to add listeners to Monoslideshow. You can do this by calling the function `addListener(event, function)` on the Monoslideshow instance. The next example instructs Monoslideshow to send the event `itemLoadComplete` to the function `onItemLoadComplete`.

```
var mss = new Monoslideshow('monoslideshowHolder');
mss.addListener('itemLoadComplete', onItemLoadComplete);
mss.load('monoslideshow.xml');

function onItemLoadComplete(evt) {
    console.log('loaded item with title ' + evt.title);
}
```

Controlling Monoslideshow via Javascript

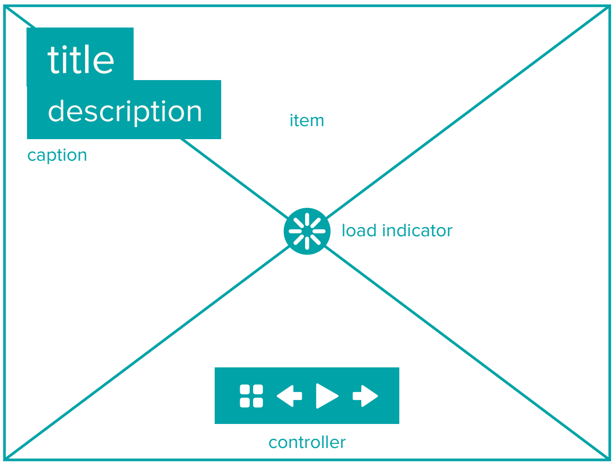
Below you find an example of how to trigger the next and previous buttons via Javascript.

```
<html>
    <head>
        <title>Monoslideshow</title>
        <script type="text/javascript" src="slideshow/monoslideshow.js"></script>
    </head>
    <body onload="onLoadComplete();">
        <div id="monoslideshowHolder" style="width: 640px; height: 480px;"></div>
        <div id="buttonPrevious">previous</div>
        <div id="buttonNext">next</div>
        <script type="text/javascript">
            function onLoadComplete() {
                var mss = new Monoslideshow('monoslideshowHolder');
                mss.load('slideshow/monoslideshow.xml');
                document.getElementById('buttonPrevious').onclick = function() {
                    mss.previous();
                }
                document.getElementById('buttonNext').onclick = function() {
                    mss.next();
                }
            }
        </script>
    </body>
</html>
```

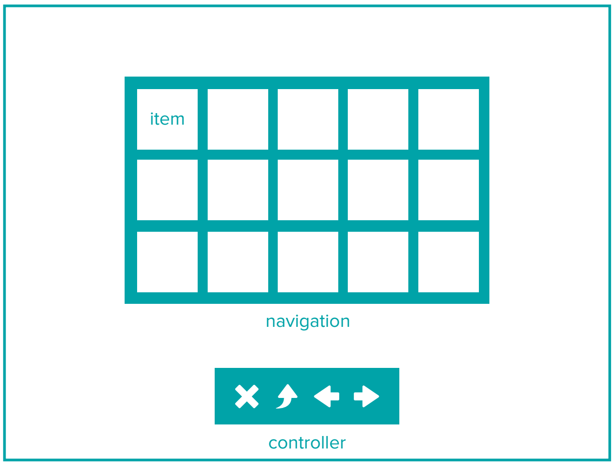
Interface elements

Interface elements

Below you see two screenshots of a typical Monoslideshow setup. The main visual objects are shown, and you can manipulate their position, size, color and a whole lot more.



This is a screenshot of the main interface. The `item` is displayed in the slideshow. There's also a `caption`, which holds the `title` and `description` of the currently displayed item. You'll see the `loadIndicator`, which indicates that an item is being loaded. And finally there's the `controller`, which lets you navigate through the slideshow.



When you click the navigation button, the slideshow pauses, the `controller` flips and the navigation window controls appear. You're now in navigation mode, and you can click on the `item` thumbnails in the `navigation` window to load a specific item. When you click on an item thumbnail, the navigation window disappears and the slideshow will resume playing.

XML reference

Album

The album element contains the contents of the slideshow. Images, videos and group elements can be used to populate an album. In addition, you can place albums inside other albums. You can specify as many items as you like, as Monoslideshow only loads items when they are needed.

```
<contents>
    <album>
        <configuration />
        <configurations />
        <itemPaths />
        <thumbnailPaths />
    </album>
</contents>
```

attribute	default	type	description
description		string	description text.
id		string	you can give an album an ID that you can use together with startWithAlbumID
itemPath		string	the path to the folder that contains all the items. the source attribute value of an item is prepended with this attribute.
thumbnail		url	the URL of a thumbnail of an album. note: this attribute is not prepended with itemPath or thumbnailPath
thumbnailPath		url	the path to the folder that contains all the thumbnails of the images. the source attribute value of an item is prepended with this attribute.
title		string	title text.

Album configuration

The album configuration element holds the configuration of an album. The global properties are defined here, such as the background color of the slideshow, the scale mode of items and what to do when the album finishes.

```
<album>
    <configuration>
        <caption />
        <controller />
        <firstTransition />
        <loadIndicator />
        <navigation />
        <pageIndicator />
        <placeholder />
        <thumbnailNavigation />
        <transition />
        <videoPlayButton />
    </configuration>
</album>
```

attribute	default	type	description
backgroundColor	#000	color/gradient	background color, can be set to transparent
groupGridHorizontalDivisions	12	number 0–MAX	horizontal divisions for the group grid
groupGridHorizontalSpacing	1	number 0–MAX	horizontal spacing between group items

attribute	default	type	description
groupGridVerticalDivisions	12	number 0–MAX	vertical divisions for the group grid
groupGridVerticalSpacing	1	number 1–MAX	vertical spacing between group items
groupItemsMax	1	number 1–MAX	when not 1, groups are randomly constructed with this value as the maximum number of items in one group
groupItemsMin	1	number 0–MAX	when not 1, groups are randomly constructed with this value as the minimum number of items in one group
delay	3	number 0–MAX	seconds of pause between items
enableEXIF	false	boolean	enable the loading of EXIF data
errorImageColor1	#111	color	background color of an error image
errorImageColor2	#333	color	foreground color of an error image
filenamePrefix		text	prepends text to all filenames, before extension
filenameSuffix		text	appends text to all filenames, before extension
onAlbumEnd	loop	option	loadNextAlbum loads next album that is at same level as current album loop loops the album showAlbum shows the album in the navigation window showParentAlbum shows the album that contains the current album(s)
randomize	false	boolean	randomizes the contents and albums of the slideshow
scaleMode	scaleToFill	option	scaleToFit scale the image so that it fits inside the viewport scaleToFill scale the image so that it fills the entire viewport none does not scale the image downscaleToFit scales the image to fit the viewport only if the resulting image has smaller dimensions than the original downscaleToFill scales the image to fill the viewport only if the resulting image has smaller dimensions than the original
startWithAlbumID		text	starts the slideshow with the album with the specified id.
startWithItemID		text	starts the slideshow with the item with the specified id.
startWithNavigationWindow	false	boolean	start with displaying the navigation window, so you can start the slideshow by selecting an item
swipeDirection	horizontal	option	horizontal swipes the slideshow horizontally vertical swipe the slideshow vertically note: use this in conjunction with the direction value of the transition element for a coherent effect.
swipeLoop	true	boolean	slideshow is endlessly swipeable.
swipeScale	0.9	number 0–1	when swiping the slideshow, the items are scaled back according to this value
viewport	0,0,0,0	offset	specifies the viewport where the items are displayed in. The format is x, y, width, height, where x, y are the top left coordinates of the viewport. You can also specify the viewport as left, top, right, bottom, "offset" (without quotes), in which case the viewport scales along with the dimensions of the div and the 4 coordinates function as margins. For example, a value of 10, 20, 30, 40, offset will place the viewport 10 pixels from the left, 20 from the top, 30 from the right and 40 from the bottom. note: a value of 0, 0, 0, 0 means a full size viewport.

Image

The image element defines an image in the slideshow.

```
<album>
    <image>
        <configuration />
        <sources />
        <thumbnails />
    </image>
</album>
```

attribute	default	type	description
data		text	this user defined string is sent together with events broadcasted by Monoslideshow
description		text	description text
id		text	you can give an image an ID that you can use together with startWithItemID in the album's configuration node
itemPath		string	the path to the folder that contains all the items. the source attribute value of an item is prepended with this attribute.
link		url	when clicked, go to this link
linkTarget		text	target frame to load the link in, for example _blank or _parent
source		url	url of the image
thumbnail		url	url to a thumbnail of this image, used in the navigation window.
thumbnailPath		url	the path to the folder that contains all the thumbnails of the images. the source attribute value of an item is prepended with this attribute.
title		text	title text
weight	1	number 0–MAX	affects the relative size of this image in group layouts

Image configuration

The image configuration element holds the configuration of an image.

```
<image>
    <configuration>
        <caption />
        <transition />
    </configuration>
</image>
```

attribute	default	type	description
delay	3	number 0–MAX	seconds of pause between images
enableEXIF	false	boolean	enable the loading of EXIF data
scaleMode	scaleToFill	option	scaleToFit scale the image so that it fits inside the viewport scaleToFill scale the image so that it fills the entire viewport none does not scale the image downscaleToFit scales the image to fit the viewport only if the resulting image has smaller dimensions than the original downscaleToFill scales the image to fill the viewport only if the resulting image has smaller dimensions than the original

Image sources variant

The image sources variant defines a variant of the same image.

```
<image>
    <sources>
        <variant />
    </sources>
</image>
```

attribute	default	type	description
height	0	number 0–MAX	height of image
source		text	url of the image variant
weight	0	number 0–MAX	width of image

Image thumbnails variant

The image thumbnails variant defines a variant of the same thumbnail.

```
<image>
    <thumbnails>
        <variant />
    </thumbnails>
</image>
```

attribute	default	type	description
height	0	number 0–MAX	height of image
source		text	url of the image variant
weight	0	number 0–MAX	width of image

Video

The video element contains a video file. Not every device supports all video formats. Please use the `sources` tag to specify multiple sources for all the devices you are targeting.

```
<album>
    <video>
        <configuration />
        <posters />
        <sources />
        <thumbnails />
    </video>
</album>
```

attribute	default	type	description
data		text	this user defined string is sent together with events broadcasted by Monoslideshow
description		text	description text
id		text	you can give an image an ID that you can use together with `startWithItemID` in the album's `configuration` node
itemPath		string	the path to the folder that contains all the items. the `source` attribute value of an item is prepended with this attribute.
link		url	when clicked, go to this link
linkTarget		text	target frame to load the `link` in, for example `_blank` or `_parent`

attribute	default	type	description
poster		text	url of a poster image for the video. Displayed when loading takes too long, and on devices that won't play inline video automatically. note: use an image with the exact dimensions of the video. The url is affected by itemPath, just like regular items.:
source		url	url of the video
thumbnail		url	url to a thumbnail of this image, used in the navigation window.
thumbnailPath		url	the path to the folder that contains all the thumbnails of the images. the source attribute value of an item is prepended with this attribute.
title		text	title text

Video configuration

The video configuration element holds the configuration of a video.

```
<video>
    <configuration>
        <caption />
        <transition />
    </configuration>
</video>
```

attribute	default	type	description
delay	3	number 0–MAX	the delay between images
scaleMode	scaleToFill	option	scaleToFit scale the image so that it fits inside the viewport scaleToFill scale the image so that it fills the entire viewport none does not scale the image downscaleToFit scales the image to fit the viewport only if the resulting image has smaller dimensions than the original downscaleToFill scales the image to fill the viewport only if the resulting image has smaller dimensions than the original
volume	1	number 0–1	the volume of the video

Video sources variant

The video sources variant specifies a variant of the same video. This is useful to target multiple browsers with different video playback capabilities..

```
<video>
    <sources>
        <variant />
    </sources>
</video>
```

attribute	default	type	description
height	0	number 0–MAX	height of video
source		text	url of the image variant
type		text	type of the video source, used to specify audio and video codecs
weight	0	number 0–MAX	width of video

Video thumbnails variant

The video thumbnails variant defines a variant of the same thumbnail.

```
<video>
    <thumbnails>
        <variant />
    </thumbnails>
</video>
```

attribute	default	type	description
height	0	number 0–MAX	height of image
source		text	url of the image variant
weight	0	number 0–MAX	width of image

Video posters variant

The video posters variant defines a variant of the same poster image of a video.

```
<video>
    <posters>
        <variant />
    </posters>
</video>
```

attribute	default	type	description
height	0	number 0–MAX	height of image
source		text	url of the image variant note: the url is affected by itemPath, just like regular items.:
weight	0	number 0–MAX	width of image

Video play button

Normally, videos are played instantly when they load. Whenever it takes too long, a video play button is displayed. The same button is also used on mobile devices that don't support automatic playback of videos.

```
<configuration>
    <videoPlayButton>
        <emboss" />
        <gloss" />
        <shadow />
    </videoPlayButton>
</configuration>
```

attribute	default	type	description
backgroundAlpha	0.95	number 0–1	background opacity of the button
backgroundColor	#222	color/gradient	background color of the button
backgroundRadius	120	number 0–MAX	roundness of the button
backgroundRadiusExponent	3	number 2–MAX	make the radius circular (2) or more rectangular (>2)
delay	1	number 0–MAX	time in seconds before the video play button will display. if the video is loaded before the time passes, the video is played automatically when supported by the device.
height	80	number 0–MAX	height of the button
iconColorDown	#cc2437	color	color of icons when clicked
iconColorOver	#ff2d46	color	color of icons when hovered over
iconColorUp	#fff	color	normal color of icons

attribute	default	type	description
margin	0	margin	margin around the button
padding	40	padding	padding inside the button
position	center	option	topCenter, topRight, rightCenter, bottomRight, bottomCenter, bottomLeft, leftCenter, topLeft, center position of the button in the slideshow
strokeColor	#000	color/gradient	color of the stroke around the button
strokeWidth	1	number 0–MAX	width of the stroke around the button
width	80	number 0–MAX	width of the button

Group

The group's contents contain multiple images to be displayed on one slide. Group contents can not contain videos or other groups.

```
<album>
    <group>
        <configuration />
        <contents />
    </group>
</album>
```

attribute	default	type	description
data		text	this user defined string is sent together with events broadcasted by Monoslideshow
description		text	description text
id		text	you can give a group an ID that you can use together with startWithItemID in the album’s configuration node
itemPath		string	the path to the folder that contains all the items. the source attribute value of an item is prepended with this attribute.
link		url	when clicked, go to this link
linkTarget		text	target frame to load the link in, for example _blank or _parent
thumbnail		url	url to a thumbnail of this image, used in the navigation window.
thumbnailPath		url	the path to the folder that contains all the thumbnails of the images. the source attribute value of an item is prepended with this attribute.
title		text	title text

Group configuration

The group configuration element holds the configuration of a group.

```
<group>
    <configuration>
        <caption />
        <transition />
    </configuration>
</group>
```

attribute	default	type	description
delay	3	number 0–MAX	seconds of pause between images
groupGridHorizontalDivisions	12	number 0–MAX	horizontal divisions of the group grid. The smart group layout determined by Monoslideshow is placed on the group grid, so there's a always a design consistency.
groupGridHorizontalSpacing	1	number 0–MAX	horizontal spacing between group items

attribute	default	type	description
groupGridVerticalDivisions	12	number 0–MAX	vertical divisions of the group grid. The smart group layout determined by Monoslideshow is placed on the group grid, so there's a always a design consistency.
groupGridVerticalSpacing	1	number 0–MAX	vertical spacing between group items
scaleMode	scaleToFill	option	scaleToFit scale the image so that it fits inside the viewport scaleToFill scale the image so that it fills the entire viewport none does not scale the image downscaleToFit scales the image to fit the viewport only if the resulting image has smaller dimensions than the original downscaleToFill scales the image to fill the viewport only if the resulting image has smaller dimensions than the original

Transition

The transition element determines which transition to perform when loading and displaying a new item. Use the tag `<firstTransition>` to define the first transition of the slideshow, which only plays once. This tag has the same syntax as the regular `<transition>` tag.

```
<configuration>
    <transition />
</configuration>
```

attribute	default	type	description
applyOnGroup	false	boolean	when true, the transition is applied on the complete group of items, instead of per group item. note: when using WebGL effects, this attribute will be forced to true.
delay	0	number 0–MAX	delay of performing the next transition
direction	leftToRight	option	topToBottom, bottomToTop, leftToRight, rightToLeft, random direction of the transition, when type is set to cube, flip, push, stack
easing	easeInOutCubic	option	easeIn, easeOut, easeInOut, easeInCubic, easeOutCubic, easeInOutCubic, none determines the easing mode during the transition
groupAppearDirection	leftToRight	option	topToBottom, bottomToTop, leftToRight, rightToLeft, random direction of the group transition
groupItemDelay	0.25	number 0–MAX	delay between group items
kenBurnsEndRectangle		rectangle	the viewing rectangle (in item space coordinates ranging from 0.0 to 1.0) to end the Ken Burns transition with note: rectangle defined as x, y, width, height. that rectangle is then scaled to fill the viewport (with mode scaleToFill).
kenBurnsMode	none	option	randomPan randomly pans accross the item randomZoomIn, randomZoomOut, randomZoom randomly zooms the item randomZoomAndPan randomly zoom or pans accross the item autoPan, autoPanBackwards always show the entire item by panning horizontally or vertically zoomIn, zoomOut zooms from the center none don't apply the Ken Burns effect note: when using group items and WebGL effects, this attribute will be forced to none.
kenBurnsStartRectangle		rectangle	the viewing rectangle (in item space coordinates ranging from 0.0 to 1.0) to start the Ken Burns transition with note: rectangle defined as x, y, width, height. that rectangle is then scaled to fill the viewport (with mode scaleToFill):

attribute	default	type	description
kenBurnsStrength	0.15	number 0–1	how much the Ken Burns effect will affect the image
kenBurnsTime	6	number 0–MAX	time in seconds the Ken burns effect is playing
pauseTransitionTime	0	number 0–MAX	the transition time of items if the slideshow is paused
performOutro	false	booelean	perform transition outro or start with intro of next item
pivotPoint	topLeft	option	topRight, bottomRight, bottomLeft, topLeft, random used if type is pivot
size	0.1	number 0–1	the size of the transition type applied (if type is focusFade, pivot, push, stack, water
strength	0.1	number 0–1	the strength of the transition type (if type is focusFade, iris, pivot, spot, stack, water, zoomIn, zoomOut
time	0.5	number 0–MAX	transition time in seconds
type	blend	option list	blend, cube, flash, flip, focusFade, grayscale, iris, noise, pivot, push, selfMelt, shatter, spot, stack, water, waterDrop, zoomIn, zoomOut CSS 3D effects: cube, flip CSS Filter effects: focusFade, grayScale WebGL effects: iris, noise, selfMelt, shatter, spot, water, waterDrop, zoomIn, zoomOut note: you can use multiple types separated by a comma in order to instruct Monoslideshow to try a different type whenever the first type is not supported.

Caption

The caption element displays a title and a description of an item. You can use your own fonts or you can use a system font. For a list of safe system fonts, you can check en.wikipedia.org/wiki/Core_fonts_for_the_Web

```
<configuration>
    <caption>
        <description />
        <emboss />
        <gloss />
        <shadow />
        <title />
        <transition />
    </caption>
</configuration>
```

attribute	default	type	description
backgroundAlpha	0.75	number 0–1	the opacity of the background
backgroundColor	#000	color/gradient	the background color of the caption element. If backgroundType is fitTextLines, gradient isn't allowed.
backgroundRadius	4	number 0–MAX	the roundness of the caption element
backgroundRadiusExponent	3	number 2–MAX	make the radius circular (2) or more rectangular (>2)
backgroundType	fitTextLines	option	fitCompleteText fits the background around the complete text field. this results in a rectangular shape. fitTextLines fits the background around the individual text lines. this results in multiple rectangular shapes.
displayMode	always	option	always always display the caption onFocus only display caption when the mouse hovers over the slideshow never never display the caption
margin	16	margin	margin around the caption element
padding	8,16	padding	padding inside the caption element
position	topLeft	option	topCenter, topRight, rightCenter, bottomRight, bottomCenter, bottomLeft, leftCenter, topLeft, center
spacing	0	number 0–MAX	spacing between the title and the description text
strokeColor	#fff	color/gradient	color of the stroke around the caption element

attribute	default	type	description
strokeAlpha	1	number 0–1	opacity of the stroke
strokeWidth	0	number 0–MAX	width of the stroke around the caption element
textAlign	left	option	left, right, center
textAlpha	1	number 0–1	opacity of the text
width	0	number 0–MAX/ percentage	width of the caption. if the value is not 0, the backgroundType is set to fitCompleteText and the caption gets a fixed width. You can also enter a percentage, which stretches the caption to the corresponding part of the slideshow window.

Caption transition

The transition element determines which transition to perform when displaying a caption.

```
<caption>
    <transition />
</caption>
```

attribute	default	type	description
direction	leftToRight	option	topToBottom, bottomToTop, leftToRight, rightToLeft used if type is moveIn
easing	easeInOutCubic	option	easeIn, easeOut, easeInOut, easeInCubic, easeOutCubic, easeInOutCubic, none determines the easing mode during the transition
time	0.5	number 0–MAX	transition time in seconds
type	blend	option	blend, moveIn

Caption title

This element holds the title configuration of a caption.

```
<caption>
    <title>
        <shadow />
    </title>
</caption>
```

attribute	default	type	description
color	#fff	color	color of the text
font	Helvetica, arial, sans-serif	string	name of a system font, or a custom defined font class loaded via fontFile
lineHeight	120%	number / %	space between text lines
letterSpacing	0	number	space between letters
size	32	number 1–MAX	font size
style	normal	option	normal, italic
template		text	template used for titles. variables are enclosed within curly brackets and are filled in automatically by built in information and EXIF data, if enableEXIF is set to true (see EXIF reference in this manual). The following built in variables are available: {fileName}, {fileBaseName}, {fileExtension}, {title} (original title), {description} (original description), {index} (current item index), {albumSize} (current album size), {albumTitle} (title of current album), {albumDescription} (description of current album). If the template attribute is not set, the caption uses the normal title, as defined in the image node. If a variable is not present (for example, not all images have all EXIF data), the variable is left out in the resulting text. If no variable in the template is found, the template is not used.

attribute	default	type	description
type	normal	option	normal, none
weight	normal	option	normal, bold

Caption description

This element holds the description configuration of a caption.

```
<caption>
    <description>
        <shadow />
    </description>
</caption>
```

attribute	default	type	description
color	#888	color	color of the text
font	Helvetica, arial, sans-serif	string	name of a system font, or a custom defined font class loaded via fontFile
lineHeight	120%	number / %	space between text lines
letterSpacing	0	number	space between letters
size	16	number 1–MAX	font size
style	normal	option	normal, italic
template		text	template used for titles. variables are enclosed within curly brackets and are filled in automatically by built in information and EXIF data, if enableEXIF is set to true (see EXIF reference in this manual). The following built in variables are available: {fileName}, {fileBaseName}, {fileExtension}, {title} (original title), {description} (original description), {index} (current item index), {albumSize} (current album size), {albumTitle} (title of current album), {albumDescription} (description of current album). If the template attribute is not set, the caption uses the normal description, as defined in the image node. If a variable is not present (for example, not all images have all EXIF data), the variable is left out in the resulting text. If no variable in the template is found, the template is not used.
type	normal	option	normal, none
weight	normal	option	normal, bold

Controller

The controller element holds the buttons to navigate through the slideshow. If the navigation window that contains the thumbnails is shown, the controller flips and shows the buttons for the navigation window.

```
<album>
    <controller>
        <gloss strength="0.25" />
        <emboss strength="0.25" />
        <scrubBar />
        <shadow />
        <transition />
    </controller>
</album>
```

attribute	default	type	description
appearTime	0.4	number 0–MAX	appear transition in seconds
autoHide	true	boolean	auto hides the controller after disappearDelay seconds
autoPlay	true	boolean	automatically plays the slideshow
backgroundAlpha	0.95	number 0–1	background opacity of the controller element

attribute	default	type	description
backgroundColor	#222	color/gradient	background color of the controller element
backgroundRadius	16	number 0–MAX	roundness of the controller element
backgroundRadiusExponent	3	number 2–MAX	make the radius circular (2) or more rectangular (>2)
disappearDelay	3	number 0–MAX	time in seconds before the controller element hides away. time starts running when there's no mouse interaction.
disappearMode	vertical	option	horizontal, vertical, fade specifies if the controller should disappear horizontally, vertically. or fade out in place
iconColorDisable	#555	color	color of icons that are disabled
iconColorDown	#cc2437	color	color of icons when clicked
iconColorOver	#ff2d46	color	color of icons when hovered over
iconColorUp	#fff	color	normal color of icons
iconSize	24	number 0–MAX	height of an icon
iconSpacing	24	number 0–MAX	spacing between icons
margin	16	margin	margin around controller element
padding	16,24	padding	padding inside the controller element
position	bottomCenter	option	topCenter, topRight, rightCenter, bottomRight, bottomCenter, bottomLeft, leftCenter, topLeft, center position of the controller element in the slideshow
resizeTime	0.4	number 0–MAX	resize transition time when the scrub bar is displayed and the controller element changes its width
showFullScreenAlbumUpButton	true	boolean	show full screen button
showNavigationAlbumUpButton	true	boolean	show album up button in the navigation window
showNavigationButton	true	boolean	show navigation button
showNavigationCloseButton	true	boolean	show close button in the navigation window
showNavigationNextButton	true	boolean	show next button in the navigation window
showNavigationPreviousButton	true	boolean	show previous button in the navigation window
showNextButton	true	boolean	show next button
showPlayButton	true	boolean	show play button
showPreviousButton	true	boolean	show previous button
showScrubBar	true	boolean	show scrub bar
startHidden	false	boolean	hides the controller at start
strokeColor	#000	color/gradient	color of the stroke around the controller element
strokeAlpha	0.75	number 0–1	opacity of the stroke
strokeWidth	21	number 0–MAX	width of the stroke around the controller element
transitionTime	0.4	number 0–MAX	show and hide transition time in seconds
type	normal	option	none do not display the controller normal display the controller

Controller transition

Determines the transition of the controller when the navigation window is shown.

```
</configuration>
    <transition />
</configuration>
```

attribute	default	type	description
direction	leftToRight	option	topToBottom, bottomToTop, leftToRight, rightToLeft used if type is flip
easing	easeInOutCubic	option	easeIn, easeOut, easeInOut, easeInCubic, easeOutCubic, easeInOutCubic, none determines the easing mode during the transition

attribute	default	type	description
time	0.4	number 0–MAX	transition time in seconds
type	flip	option	blend, flip

Controller scrub bar

The scrub bar element will expand when a video is displayed. You can use the scrub bar to navigate through the timeline of a video.

```
</configuration>
    <scrubBar />
</configuration>
```

attribute	default	type	description
backgroundColor	#888	color	background color
height	8	number 0–MAX	height
interactionEnabled	true	boolean	determines if the scrub bar has mouse interaction
loadingColor	#fff	color	color of the loader indicator
margin	0	margin	margin around the element
progressColor	#ff2d46	color	color of the progress indicator
width	160	number 0–MAX	width

Side buttons

The side buttons element holds the previous / next buttons to navigate through the slideshow. It can be used as a simpler alternative to the controller element.

```
<album>
    <sideButtons>
        <gloss strength="0.25" />
        <emboss strength="0.25" />
        <shadow />
    </sideButtons>
</album>
```

attribute	default	type	description
appearTime	0.4	number 0–MAX	appear transition in seconds
autoHide	true	boolean	auto hides the controller after disappearDelay seconds
backgroundAlpha	0.95	number 0–1	background opacity of the controller element
backgroundColor	#222	color/gradient	background color of the controller element
backgroundRadius	16	number 0–MAX	roundness of the controller element
backgroundRadiusExponent	3	number 2–MAX	make the radius circular (2) or more rectangular (>2)
disappearDelay	3	number 0–MAX	time in seconds before the controller element hides away. time starts running when there's no mouse interaction.
height	24	number 0–MAX	height of an icon
iconColorDown	#cc2437	color	color of icons when clicked
iconColorOver	#ff2d46	color	color of icons when hovered over
iconColorUp	#fff	color	normal color of icons
margin	16	margin	margin around buttons
padding	12	padding	padding inside the buttons
position	center	option	top, bottom, left, right, center position of the controller element in the slideshow
startHidden	false	boolean	hides the controller at start
strokeColor	#000	color/gradient	color of the stroke around the controller element
strokeAlpha	0.75	number 0–1	opacity of the stroke

attribute	default	type	description
strokeWidth	21	number 0–MAX	width of the stroke around the controller element
transitionTime	0.4	number 0–MAX	show and hide transition time in seconds
type	normal	option	none do not display the controller horizontal, vertical display the controller
width	24	number 0–MAX	width of an icon

Navigation

The navigation element displays the thumbnails of the contents of an album. You can navigate to sub albums and select items to display.

```
<configuration>
    <navigation>
        <albums />
        <caption />
        <items />
        <pageIndicator type="horizontal" />
        <shadow />
        <gloss />
        <emboss />
    </navigation>
</configuration>
```

attribute	default	type	description
backgroundOverlayAlpha	0.5	number 0–1	the opacity of the overlay behind the navigation
backgroundOverlayColor	#000	color/gradient	the color of the overlay behind the navigation
backgroundOverlayFadeInTime	0.3	number 0–MAX	the fade in time of the overlay behind the navigation
backgroundAlpha	0.75	number 0–1	opacity of background
backgroundColor	#000	color/gradient	color of background
backgroundRadius	16	number 0–MAX	roundness of background
backgroundRadiusExponent	3	number 2–MAX	make the radius circular (2) or more rectangular (>2)
margin	88	margin	space around element
errorImageColor1	#111	color	background color of an error image
errorImageColor2	#333	color	foreground color of an error image
padding	24	padding	space inside element
position	center	option	topCenter, topRight, rightCenter, bottomRight, bottomCenter, bottomLeft, leftCenter, topLeft, center
strokeColor	#fff	color/gradient	color of line around element
strokeAlpha	1	number 0–1	opacity of the stroke
strokeWidth	0	number 0–MAX	width of line around element

Navigation caption

The navigation caption element displays the title of the album of which the contents are being displayed.

```
<navigation>
    <caption>
        <description />
        <title />
    </caption>
</navigation>
```

attribute	default	type	description
albumTitlePlaceholder	…	string	in sub albums, place this in front of the title, to signify that the current album is placed inside other albums

attribute	default	type	description
albumTitleSeparator	/	string	in sub albums, separate album titles with this string
backgroundAlpha	0.75	number 0–1	the opacity of the background color
backgroundColor	#000	color/gradient	the background color of the caption element
backgroundRadius	4	number 0–MAX	the roundness of the caption element
backgroundRadiusExponent	3	number 2–MAX	make the radius circular (2) or more rectangular (>2)
backgroundType	fitTextLines	option	fitCompleteText fits the background around the complete text field. this results in a rectangular shape. fitTextLines fits the background around the individual text lines. this results in multiple rectangular shapes.
displayFullTitles	false	boolean	in sub albums, display full titles, or, if false, display albumTitlePlaceholder
displayParentAlbums	true	boolean	in sub albums, display parent albums in front of the current title
margin	16	margin	margin around the caption element
padding	8,16	padding	padding inside the caption element
position	left	option	left, right, center
spacing	0	number 0–MAX	spacing between the title and the description text
strokeColor	#fff	color/gradient	color of the stroke around the caption element
strokeAlpha	1	number 0–1	opacity of the stroke
strokeWidth	0		width of the stroke around the caption element
textAlign	left	option	left, right, center
textAlpha	1	number 0–1	opacity of the text
type	normal	option	normal display caption none do not display caption

Navigation items

This element defines the characteristics of item thumbnails in the navigation window.

```
<navigation>
    <items>
        <caption />
        <emboss />
        <gloss />
        <imageEmboss />
        <imageGloss />
        <imageShadow />
        <placeholder />
        <shadow />
        <transition />
    </items>
</navigation>
```

attribute	default	type	description
backgroundAlpha	1	number 0–1	the opacity of the background color
backgroundColor	#222	color/gradient	the background color of an image
backgroundHighlightAlpha	0.1	number 0–1	the opacity of the background highlight color
backgroundHighlightColor	#fff	color/gradient	the background highlight color of the image
backgroundRadius	4	number 0–MAX	the roundness of an image
backgroundRadiusExponent	3	number 2–MAX	make the radius circular (2) or more rectangular (>2)
columns	0	number 0–MAX	the number of columns note: use a value of 0 to let Monoslideshow calculate the optimal number of columns
fadeInTime	0.25	number 0–MAX	album fade in time in seconds
groupGridHorizontalSpacing	1	number 0–MAX	horizontal spacing between group items

attribute	default	type	description
groupGridVerticalSpacing	1	number 0–MAX	vertical spacing between group items
height	80	number 0–MAX	height of the image
horizontalSpacing	2	number 0–MAX	horizontal space between images
imageBackgroundAlpha	1	number 0–1	background opacity of the thumbnail image
imageBackgroundColor	#000	color/gradient	background color of the thumbnail image
imageRadius	4	number 0–MAX	roundness of the thumbnail image
imageRadiusExponent	3	number 2–MAX	make the radius circular (2) or more rectangular (>2)
imageScaleMode	scaleToFill	option	scaleToFit, scaleToFill, noScale, downscaleToFit, downscaleToFill the scale mode of the thumbnail image
imageStrokeAlpha	1	number 0–1	opacity of the stroke around the image
imageStrokColor	#000	color/gradient	color of the stroke around the image
imageStrokeWidth	0	number 0–MAX	width of the stroke around the image
padding	0	padding	padding inside the image
rows	0	number 0–MAX	the number of rows. note: use a value of 0 to let Monoslideshow calculate the optimal number of rows
strokeAlpha	1	number 0–1	opacity of the stroke around the image
strokeColor	#000	color/gradient	color of the stroke around the image
strokeWidth	0	number 0–MAX	width of the stroke around the image
verticalSpacing	2	number 0–MAX	vertical space between image
width	80	number 0–MAX	width of the image

Navigation item caption

The album caption element of an album in the navigation window.

```
<*>
    <caption>
        <description size="10" />
        <title size="12" />
    </caption>
</*>
```

attribute	default	type	description
backgroundAlpha	0.75	number 0–1	the opacity of the background color
backgroundColor	#000	color/gradient	the background color of the caption element
backgroundRadius	4	number 0–MAX	the roundness of the caption element
backgroundRadiusExponent	3	number 2–MAX	make the radius circular (2) or more rectangular (>2)
backgroundType	fitTextLines	option	fitCompleteText fits the background around the complete text field. this results in a rectangular shape. fitTextLines fits the background around the individual text lines. this results in multiple rectangular shapes.
height	64	number 0–MAX	height of the caption. note: a value of 0 is for autosizing the height based on the width
margin	16	margin	margin around the caption element
padding	8,16	padding	padding inside the caption element
spacing	0	number 0–MAX	spacing between the title and the description text
strokeColor	#fff	color/gradient	color of the stroke around the caption element
strokeAlpha	1	number 0–1	opacity of the stroke around the caption element
strokeWidth	0	numver 0–MAX	width of the stroke around the caption element
textAlign	left	option	left, right, center
textAlpha	1	number 0–1	opacity of the text
width	128	number 0–MAX	width of the caption

Navigation item transition

This element defines the transition type when hovering over an album or item in the navigation window.

```
<*>
    <transition>
        <imageGloss strength="0.4" />
        <imageShadow size="16" />
        <shadow size="16" />
    </transition>
</*>
```

attribute	default	type	description
distanceX	0	number MIN-MAX	hover horizontal translation
distanceY	0	number MIN-MAX	hover vertical translation
easing	easeInOutCubic	option	easeIn, easeOut, easeInOut, easeInCubic, easeOutCubic, easeInOutCubic, none determines the easing mode during the transition
scale	1.2	number 1-2	scale factor when hovering over the element
time	0.1	number 0-MAX	highlight transition time in seconds
type	normal	option	normal, none hover highlight mode

Navigation albums

The navigation album configuration element holds the configuration of albums in the navigation window. An album contains a thumbnail and a caption.

```
<navigation>
    <albums>
        <caption type="normal" margin="8,0,0,0" />
        <emboss />
        <gloss />
        <imageEmboss />
        <imageGloss strength="0" roundness="0.1" />
        <imageShadow size="0" alpha="0" distance="0" />
        <placeholder />
        <shadow size="16" distance="4" />
        <transition scale="1" distanceY="-5">
            <imageGloss strength="0" />
            <imageShadow size="0" alpha="0" distance="0" />
            <shadow distance="4" />
        </transition>
    </albums>
</navigation>
```

attribute	default	type	description
backgroundAlpha	1	number 0-1	the opacity of the background color
backgroundColor	#222	color/gradient	the background color of an album
backgroundHighlightAlpha	0.1	number 0-1	the opacity of the background highlight color
backgroundHighlightColor	#fff	color/gradient	the background highlight color of an album
backgroundRadius	4	number 0-MAX	the roundness of an album
backgroundRadiusExponent	3	number 2-MAX	make the radius circular (2) or more rectangular (>2)
columns	0	number 0-MAX	the number of columns note: use a value of 0 to let Monoslideshow calculate the optimal number of columns
fadeInTime	0.25	number 0-MAX	album fade in time in seconds
groupGridHorizontalSpacing	1	number 0-MAX	horizontal spacing between group items
groupGridVerticalSpacing	1	number 0-MAX	vertical spacing between group items
height	96	number 0-MAX	height of an album element

attribute	default	type	description
horizontalSpacing	8	number 0–MAX	horizontal space between albums
imageBackgroundAlpha	1	number 0–1	background opacity of the thumbnail image in an album
imageBackgroundColor	#000	color/gradient	background color of the thumbnail image in an album
imageRadius	4	number 0–MAX	roundness of the thumbnail image in an album
imageRadiusExponent	3	number 2–MAX	make the radius circular (2) or more rectangular (>2)
imageScaleMode	scaleToFill	option	scaleToFit, scaleToFill, noScale, downscaleToFit, downscaleToFill the scale mode of the thumbnail image in an album
imageStrokeAlpha	0.25	number 0–1	opacity of the stroke around the image
imageStrokColor	#000	color/gradient	color of the stroke around the image
imageStrokeWidth	1	number 0–MAX	width of the stroke around the image
padding	20	padding	padding inside the album
rows	0	number 0–MAX	the number of rows. note: use a value of 0 to let Monoslideshow calculate the optimal number of rows
strokeAlpha	1	number 0–1	opacity of the stroke around the album
strokeColor	#000	color/gradient	color of the stroke around the album
strokeWidth	1	number 0–MAX	width of the stroke around the album
verticalSpacing	8	number 0–MAX	vertical space between albums
width	128	number 0–MAX	width of an album

Thumbnail navigation

The thumbnail navigation window is a permanent navigation window that only contains items and a previous and next button.

```
<configuration>
    <thumbnailNavigation>
        <caption />
        <items rows="1" width="48" height="48" imageRadius="4" />
        <pageIndicator />
        <shadow />
        <gloss />
        <emboss />
    </thumbnailNavigation>
</configuration>
```

attribute	default	type	description
autoFollow	true	boolean	current thumbnail ishighlighted as the slideshow plays
backgroundAlpha	0.75	number 0–1	opacity of background
backgroundColor	#000	color/gradient	color of background
backgroundRadius	8	number 0–MAX	roundness of background
backgroundRadiusExponent	3	number 2–MAX	make the radius circular (2) or more rectangular (>2)
margin	16	margin	space around element
iconColorDisable	#555	color	color of icons that are disabled
iconColorDown	#cc2437	color	color of icons when clicked
iconColorOver	#ff2d46	color	color of icons when hovered over
iconColorUp	#fff	color	normal color of icons
iconMargin	8	margin	margin around the icons
iconMode	horizontal	option	horizontal, vertical
iconSize	24	number 0–MAX	size of the icons
padding	8,16	padding	space inside element
position	bottomCenter	option	topCenter, topRight, rightCenter, bottomRight, bottomCenter, bottomLeft, leftCenter, topLeft, center
showNextButton	true	boolean	shows the next button
showPreviousButton	true	boolean	shows the previous button
strokeColor	#fff	color/gradient	color of line around element
strokeAlpha	1	number 0–1	opacity of the stroke
strokeWidth	0	number 0–MAX	width of line around element

attribute	default	type	description
strokeWidth	0	number 0–MAX	width of line around element
swipeDirection	horizontal	option	horizontal swipes the navigation window horizontally vertical swipe the navigation window vertically note: use this in conjunction with the iconMode value for a coherent effect.
swipeLoop	true	boolean	navigation window is endlessly swipeable.
swipeScale	0.9	number 0–1	when swiping the navigation window, the items are scaled back according to this value
type	none	option	normal display this element none don't display the element

Load indicator

The load indicator element is a spinner that's displayed when items take long to load.

```
<configuration>
    <loadIndicator>
        <emboss />
        <gloss strength="0" />
        <shadow strength="0" />
    </loadIndicator>
</configuration>
```

attribute	default	type	description
backgroundAlpha	0.75	number 0–1	opacity of the background
backgroundColor	#000000	color	color of the background
backgroundRadius	16	number 0–MAX	roundness of background
backgroundRadiusExponent	3	number 2–MAX	make the radius circular (2) or more rectangular (>2)
delay	1.5	number 0–MAX	delay in seconds before load indicator is shown
lineColor	#333	color	color of the lines
lineColorActive	#fff	color	active color of the lines
lineLength	10	number 0–MAX	length of lines
lineRadius	2	number 0–MAX	radius of line border
lineRadiusExponent	3	number 2–MAX	make the radius circular (2) or more rectangular (>2)
lines	10	number 1–MAX	number of lines
lineWidth	4	number 1–MAX	width of lines
margin	16	margin	margin around the element
padding	16	padding	padding inside the progress element
position	center	option	topCenter, topRight, rightCenter, bottomRight, bottomCenter, bottomLeft, leftCenter, topLeft, center
rotationTime	1	number 0–MAX	time in seconds for a full rotation
startRadius	20	number 1–MAX	starting radius of the lines
strokeAlpha	1	color	opacity of line around element
strokeColor	#000	color	color of line around element
strokeWidth	2	number 0–MAX	width of line around element
trailSize	0.5	number 0–1	how much the lines should have a fading trail
type	normal	option	normal display element none don't display element

Page indicator

The page indicator shows how many pages are in the navigation window and which one is currently active.

```
<*>
    <pageIndicator />
</*>
```

attribute	default	type	description
autoHide	true	boolean	auto hides if all items are visible
color	#333	color	color of background
colorAlpha	1	number 0–1	opacity of the indicator
colorActive	#fff	color	color of the active state indicator
colorActiveAlpha	1	number 0–1	opacity of the active state indicator
margin	16,16,0,16	margin	margin around element
position	bottom	option	top, borrom, left, right
radius	4	number 0–MAX	radius of the indicator
radiusExponent	2	number 2–MAX	make the radius circular (2) or more rectangular (>2)
size	8	number 0–MAX	icon width
spacing	8	number 0–MAX	space around element
strokeColor	#000	color	color of stroke
strokeAlpha	1	number 0–1	opacity of stroke
strokeWidth	0	number 0–MAX	width of line around indicator
strokeActiveColor	#000	color	color of stroke of active indicator
strokeActiveAlpha	1	number 0–1	opacity of stroke of active indicator
type	none	option	horizontal, vertical display element none don’t display element

Placeholder

The placeholder is shown when an item has not loaded yet.

```
<*>
    <placeholder>
        <emboss strength="0" />
        <gloss strength="0" />
    </placeholder>
</*>
```

attribute	default	type	description
color	#222	color	color of background
strokeColor	#000	color	color of stroke
strokeWidth	2	number 0–MAX	width of line around element

Shadow element

The shadow element can be applied to various elements. You can disable the shadow element by setting the `alpha` to `0`.

```
<*>
    <shadow />
</*>
```

attribute	default	type	description
alpha	0.5	number 0–1	opacity of the effect
angle	90	number 0–360	angle
color	#000	color	color of the effect
distance	2	number 0–MAX	distance
size	4	number 1–100	size of the shadow

Gloss element

The gloss element can be applied to various elements. You can disable the gloss element by setting the `strength` to `0`.

```
<*>
    <gloss />
</*>
```

attribute	default	type	description
color	#fff	color	color of the effect
height	0.5	number 0–1	relative height of the gloss
roundness	0.25	number 0–1	roundness of the gloss
strength	0.1	number 0–1	strength of the gloss

Emboss element

The emboss element can be applied to various elements. You can disable the emboss element by setting the `strength` to `0`.

```
<*>
    <emboss />
</*>
```

attribute	default	type	description
angle	90	number 0–360	angle
size	1	number 0–MAX	size of the emboss
strength	0.1	number 0–1	strength of the emboss

Appendix

Color and gradient syntax

Colors are defined in hex, and are always preceded by a `#`, like this: `#rrggbb`. For some elements the color can be of type `color/gradient`, which means that besides the normal color syntax, it's also possible to apply a gradient. The syntax for defining a gradient is as follows: `type,degrees-color,ratio,alpha-color,ratio,alpha`. The triplet `color,ratio,alpha` can be repeated more than two times. The values of the components are defined as:

• type	`linear` or `radial`
• degrees	degrees (0—360)
• color	any color, in hex format: #rrggbb
• ratio	the position in the total gradient at which the color stop is placed (0—1)
• alpha	the opacity of the color stop (0—1), this value is optional and by default set to 1

This example fills an element with a vertical fade from red to green: `linear,90-#ff0000,0-#00ff00,1`.

EXIF reference

The following table lists the EXIF attributes you can use in the caption template system of Monoslideshow. Please note that they are case-sensitive and all start with an uppercase character (instead of the native Monoslideshow attributes). The list is partly based on the EXIF 2.2 specification, which you can read more about here: http://www.exif.org

attribute	description
`ImageWidth`	image width
`ImageLength`	image height
`ImageDescription`	image title
`Make`	manufacturer of image input
`Model`	model of image input equipment
`Orientation`	orientation of image
`XResolution`	image resolution in width direction
`YResolution`	image resolution in height direction
`ResolutionUnit`	unit of X and Y resolution
`Software`	software used
`DateTime`	file change date and time
`Artist`	person who created the image
`Copyright`	copyright holder
`ExposureTime`	exposure time
`FNumber`	F number
`ISOSpeedRatings`	ISO speed ratings
`DateTimeOriginal`	date and time original image was generated
`DateTimeDigitized`	date and time image was made into digital data
`ShutterSpeedValue`	shutter speed
`ApertureValue`	aperture
`BrightnessValue`	brightness
`ExposureBiasValue`	exposure bias
`MaxApertureValue`	maximum lens aperture
`SubjectDistance`	subject distance
`FocalLength`	lens focal length
`MakerNote`	manufacturer note
`UserComment`	use comment

attribute	description
`ColorSpace`	color space information
`PixelXDimension`	valid image width
`PixelYDimension`	valid image height
`RelatedSoundFile`	related audio file
`DigitalZoomRatio`	digital zoom ratio
`FocalLengthIn35mmFilm`	focal length in 35mm film
`GPSLatitudeRef`	north or south latitude
`GPSLatitude`	latitude
`GPSLongitudeRef`	east or west longitude
`GPSLongitude`	longitude
`GPSAltitudeRef`	altitude reference
`GPSAltitude`	altitude
`GPSTimeStamp`	GPS time (atomic clock)
`GPSDateStamp`	GPS date