



This page contains information about how to set up the PageFlip, how to create the content, and customizing. All the XML properties are detailed. Please read it carefully before setting up a new book. If something is not covered by this documentation, please let us know.

Documentation for the previous version PageFlip 3.5
XML configuration for the version PageFlip 3.4

PageFlip 3.6 documentation

PageFlip documentation writing still in progress!

[Demo Books](#)

[Products](#)

[Documentation](#)

[Version History](#)

[Free Version](#)

Content:

New features in PageFlip 3.6

Quick Guide how to build a PageFlip

Detailed Description

- What is PageFlip?

- PageFlip main features

- PageFlip additional features

- Developer features

How to build a PageFlip

- Main parts

- Rendering preferences

- PageFlip sounds

Designing the content

- Customising the control bar

- Customising the Stage

- Load PageFlip into your own Movie

Setting up the XML file

Using Developer features

New features in version 3.6

added Background and ControlBar button colour setup to XML

totally rewritten Thumbnail scripts including display, layout, scaling, and thumbnail loading

added Always Opened feature

added support for odd number of pages

added page number display in Thumbnail mode

added ZoomFile/PrintFile to XML to control zooming/printing independently

added VerticalMode feature

added Page Offset parameter

added CopyrightText attribute to the XML language elements, no need to edit .fla anymore

added PageName feature (Pages can have names instead of a number, and the number of the page can be changed also)

added ZoomToTop switch (In Dragzoom mode the top of the zoom image is displayed first)

fixed zoom window
fixed zoom image position when resizing the Stage while zooming
fixed hotspot in zoom mode
added fullscreen button
added multipage TOC support
added optional data to each page, which id defined in XML and can be accessed from the page's script.

Review all changes made to PageFlip in the version history

Quick Guide how to build a PageFlip Book

Download the sample XML file. (IE/FF: Right click > save link as – Safari: alt click)

First set the PageWidth/PageHeight (in pixels) in the XML file, then make your page files using the same dimensions. Save them in jpg/gif/png or swf format. Set the filenames in XML to each page.

You are ready to start PageFlip.

To meet all the features and possibilities with PageFlip, read the detailed description below.

Detailed Description

What is PageFlip?

PageFlip is an application that simulates a book on your screen, let you flip pages by dragging them, or by using control buttons. Usually it is used online, embedded into a website, to present book-like content.

As PageFlip is an Adobe Flash application, written in ActionScrip 2.0 language, it have all the advantages of a Flash document, every page content can be not only a graphic symbol (picture, static vector based shapes) but an animation (frame by frame, or scripted) or interactive flash content (forms, sound/videoplayer, any flash application).

In addition PageFlip have many features that can help browsing and building content.

PageFlip main features

Pages	Each page is a single file, that can be an image, or even a Flash Movie. Can have animations, sounds, embedded video, or any interactive content. You can set up to 9998 pages in a PageFlip.
Page cache	Only a fixed number of pages are pre-loaded.
Unload pages	Unused pages can be unloaded.
Transparency	One level transparency for the top page.
Flipping pages manually	Pages can flipped by a simple Mouse gesture, by clicking on their corner or by dragging them at their outer side.
Flipping pages by control bar's buttons	Embedded control bar for navigation (back/forward, zoom, print, thumbnail mode, auto flip).
Tear off pages	Pages can be Tear off (if enabled for that page).
Animation on the pages	Frame by frame or scripted animation, and any interaction is allowed.
Zoom view	Display a closer view of the page.
Printing page	PageFlip can print the selected page if it have an associated zoom file.

Thumbnail view	Two different type of thumbnail can be used.
HotSpots	User defined shapes on the page, which displays a popup message on mouseover, and can be clicked to go to another page or URL.

PageFlip additional features

Scalable content	Content is scaled as the Stage is scaled.
AutoSize pages	Can load different sized content to the pages, and resizes them to fit the page.
Sound on page	Usage of a sound object on a page is demonstrated with a sample file (demo_sound fla). It fades the sound on and off depending on the pages visibility. An additional sample file (demo_sound2 fla) is using a special sound effect (made by http://www.webmusic.hu).
Bitmap smoothing	Page flipping and scaled bitmap pages looks better with bitmap smoothing. It is now applied to all image file loaded as page-file.
AutoFlip	"Slideshow", flip through the book automatically, with optional looping.
Right to left	Support for arabic (or any right to left written language) content.
Vertical Mode	Use for calendar like flipping.
LightBox Zoom	Zoom image is displayed using LightBox2. In case if swf Zoom File is used, the Zoom will be displayed by PageFlip.

Developer features

Page scripting	If you want more than static pages, you will need some scripting, to let pageflip control your content.
TOC (Table of contents)	This sample page demonstrates with a short script, how you can build a Table of Contents page using Data stored in the XML configuration file. It also give example how to display it on the page, making every item clickable and adding them a link to the referred page.
Sound demo page	Example how to use sound on a page. It controls the sound so it plays only if the container page is visible.
Video embed page	The same kind of page, but it controls a streamed flv playback, using VideoPlayer Flash component.
Spread Animation page	It is the most complicated type of content. A Spread in a single file. This spread is loaded twice by PageFlip, one for the left, and one for the right side. The script in the spread file have to determine, that the loaded instance from itself is on which side, and which instance is loaded first (the second will be the master). Every change on the pages must be repeated on the other side.
Animation page	As the loaded file can be an swf file, it can have any type of flash animation, or scripted interactive content.
Event Handlers	Event handlers are for adding special script to control content management, or to add visual/audio effects. The following events are watched: <ul style="list-style-type: none"> ■ start flipping

- end flipping
- tear off a page
- switch into zoom mode and back
- switch into thumbnail view mode and back

How to build a PageFlip

Before building a PageFlip, it is important to know what features would you like to use. The size of the pages, its format, the need of the thumbnail or zoom files, all depends on what you set up in XML.

Main parts

Page files

PageFlip is composed of page files. They could be jpg, gif (non animated), png or swf files. Image files are for static content, the best performance of the flip animation speed is achieved by using image files. (You can make transparent page not only by swf, but by using png format, and still keep the advantage of using an image file)

Swf files are treated specially, because they are special. They are scalable, they can be animated and scripted.

It is recommended to add the following script to each page's root:

```
function reset(onTop, active) {
//custom script here
}
```

Custom scripts have to control

- enable/disable buttons (needed when using transparent pages)
- enable/disable any interaction on the page which is not needed when the page is not visible
- sound/video playback

The variable `onTop` is true only when the page is on the top of all other. Buttons have to be enabled only in this case.

`active` is true, when the page is visible, but not on top (when the top page is transparent, or in thumbnail mode). It have to control animation on the page.

If you want to use the same swf file for the normal page view, and for the zoom view too, or you would like to use AutoSize feature, then you have to define some variables on the same level as the previous script, on the root of the page file:

```
// used for autosize;
pageWidth = 330;
pageHeight = 380;
// if used as zoom file, use these size parameters
zoomWidth = 660;
zoomHeight = 760;
zoomScale = 200;
```

The values for AutoSize tells the script how big is the loaded page's original size. Values for zoom tells the PageFlip how many it have to enlarge the content.

The PageFlip package contains a page template source file (Source/pages/template_page.fla) which contains all these functions and variables predefined. You just need to set values for the variables that match you setup.

Zoom files

Zoom files are used for a detailed page view, and printing source for a page.

Thumbnail files

If [LiveThumbnails](#) is set to **"true"** in the XML file, you don't need to make thumbnail files, as they are scaled down from the page file. Otherwise, the thumbnail files have to be the scaled down copy of the original page content.

Rendering preferences

Transparency

Enable Transparency in the XML file (Transparency="true"), to be able to use transparent content. This enables a one level page transparency, that means you can view through the page, you can view the page overlaid, but not the next. Only 1 layer is rendered in addition. This is for best performance, too much page content rendering can slow down the flow of the flip control.

If you don't want to use transparent pages, set this attribute to false (Transparency="false"). Rendering will be smoother this way.

Hardcover/Hardpage(Settings) HardPage(PageOrder/PageData)

Pages can be rendered like "hard" book covers. 3 attributes controls this option:

[Hardcover](#) (Settings' attribute)

- set to **"false"** to use "soft" cover (default value).
- set to **"true"** to render "hard" book cover.

[Hardpage](#) (Settings' attribute)

- set to **"false"** to use "soft" pages (default value).
- set to **"true"** to render inside pages as "hard" page.

[HardPage](#) (PageOrder/PageData's attribute)

- set to **"true"** to render that page as "hard" page (default false).

If you don't specify an attribute, its default value will be used!

LargeCover

As all hard-covered book's cover larger than the inside pages, PageFlip let you the option to use Larger cover pages. This feature works correctly only if you enabled Hardcover.

The following attributes are for setting up PageFlip's "Large"cover:

[LargeCover](#) (Settings' attribute)

- Enable LargeCover to use bigger cover pages than inside pages ([LargeCover](#)=**"true"**, default false).

[LargeCoverTopOversize](#) (Settings' attribute)

- Difference in pixel at top, and bottom.

[LargeCoverSideOversize](#) (Settings' attribute)

- Difference in pixel at the outer (not center) side of the cover page.

PageScale

Pages can be scaled up/down as the stage is resized. The zoom-page viewer area and the Thumbnails area are also scaled if this option is enabled (PageScale="true", default false).

The scaling is done by the following way:

PageFlip is rendered at 100% (with the page size set in the XML) only when it is displayed with same Stage size (at embed level in html, or in Flash Player) as it is defined in XML in [BaseStageWidth](#) and [BaseStageHeight](#) attributes. When the stage is resized, PageFlip fits itself inside the space left. Space left means, the [Stage.height](#) - [DefaultControlHeight](#). [DefaultControlHeight](#) is set in pageflip_loader fla, and it is the Height of the control-bar in pixels.

LiveThumbnails

Thumbnails are displayed only if [ThumbnailsEnabled](#) (Settings' attribute) is set to true ([ThumbnailsEnabled](#)=**"true"**)

[LiveThumbnails](#) means, that the thumbnail images of the pages are the pages itself, just resized. All pages are shown exactly as they appears in the book. With [ThumbnailsActivePages](#) enabled ([ThumbnailsActivePages="true"](#), default false) animation are rendered too. [LiveThumbnails](#) are rendered on stage so they fit the book's area, like they are sit on a spread.

Disable [LiveThumbnails](#) ([LiveThumbnails="false"](#), default true) to allow pageflip to load pre-saved thumbnail pages. They will be displayed with predefined fixed size, and they won't be never scaled.

The size of the loaded thumbnail images must be set (only if [LiveThumbnails](#) is enabled) in [ThumbnailWidth/ThumbnailHeight](#) (Settings' attribute).

[ThumbnailPadding](#) (Settings' attribute)

- space between thumbnail spreads in pixel.

[ThumbnailBorderWidth](#) (Settings' attribute)

- width of the border outside of the spreads.

[ThumbnailBorderColor](#) (Settings' attribute)

- the colour of that border (example "[0x333333](#)", Hexadecimal RGB value).

[ThumbnailBorderHoverColor](#) (Settings' attribute)

- the colour of that border when hovered by the pointer.

HotSpot

HotSpots are used to make special buttons on the page. They are purely defined in XML. HotSpots are invisible Shapes on the page, which displays a popup message (also set in XML) on mouse over, and can be set up to open a webpage in the browser, or to go to any page in the book.

Each page can have its own HotSpot XML file, defined in [HotspotData](#) (an optional attribute of PageOrder/PageData). Assign it the file which contains the HotSpot definitions for that page ([HotspotData="pages/page1hotspot.xml"](#)).

HotSpots are defined in an XML file. Each page with HotSpots on it, have its own HotSpot data file. The root node is [HotSpotData](#), it contains the HotSpot definitions, each within a [HotSpot](#) childnode, which have these attributes to set up a HotSpot:

[Shape](#) (HotSpot's attribute)

- Describes a free shape, point by point, using a coordinate system with origin in top-left corner of the page. The last point is connected to the first automatically. The syntax is: [Shape="x1,y1/x2,y2/x3,y3/xn,yn"](#)

[PopupTitle](#) (HotSpot's attribute)

- Popup window title, which appears on mouseover. Syntax: [PopupTitle="Your Title"](#)

[PopupText](#) (HotSpot's attribute)

- Popup window message, which appears on mouseover. Syntax: [PopupText="Your message"](#)

[TargetLabel](#) (HotSpot's attribute)

- You can go to any page with a Label (See PageOrder, PageData). Syntax: [TargetLabel="YourLabel"](#)

[Skip](#) (HotSpot's attribute)

- If true, it goes directly to the target page, if false, it goes there by flipping page by page. Syntax: [Skip="true"](#)

[Link](#) (HotSpot's attribute)

- Opens an URL in your browser. Syntax: [Link="http://pageflip.hu"](#)

Use the original HotSpot XML file as template.

PageFlip sounds

PageFlip have built in pageflip sounds, that can be controlled from XML. You can enable/disable PageFlip sounds (Sounds Enabled="true"/"false", default false), you can set up the flipping sound level from 0–100 (EffectVolume="100", default 25), and you can set custom mp3 files for 3 event: Start flip, End flip (sounds when mouse is released) and for TearOff.

Designing the content

the size of the book

The size of your page depends on the usage of the book:

- Fixed size – the book have a fixed size, all pages need to be created with the same (predefined) dimension. There is still an exception: if you set `AutoSize="true"` in the XML file, the page will be resized to the predefined size.
- Scalable – if you want your book to be scalable, you have to set the base dimension depending on the type of the pages. If you use bitmap content, the base dimension of the page have to be the largest supposed page size. But if you use mostly swf (vector) page content, there is no need to use big pages, they will be able to scaled up without quality losses.

It is important, that if you already know the page size, you have to set the `BaseStageWidth` and `BaseStageHeight` values in XML so the pages will have enough space to fit on the Stage. You have you have to be care about to embed `pageflip_loader.swf` with the same size to view pages at 100%. Otherwise, if `PageScale` is enabled, and the embedded Movie's size is not the same as `BaseStageWidth`/`BaseStageHeight`, the PageFlip will be scaled.

`BaseStageWidth`/`BaseStageHeight` is used to calculate the scaling factor of the book. These variables and the Stage size must be large enough to display the book and the controls on the Stage without cropping.

Example:

If you want to use 400x600 sized pages, your book will be 800x600, add then the controlbar height -> 800x660, and add some space at the top and bottom of the book -> 800x800

Set your Stage size to 800x800 pixels (this step can be skipped if you embed PageFlip into html)

Set these attributes in XML:

`BaseStageWidth="800"`

`BaseStageHeight="800"`

Set your Movie size in the embed scripts in html (this step can be skipped if run PageFlip in StandAlone Flash Player)

Now your pages will be displayed as expected.

pages

Page files must be made using the dimensions set in XML (`PageWidth`/`PageHeight` Settings' attributes), except if you enable `AutoSize` option in XML (`AutoSize` Settings' attribute, default true), but better is the quality with matching size.

Buttons on the page

Making buttons needs a bit more attention than non-clickable objects. If transparency is enabled/used, a page can still visible even if it isn't on top. if there are buttons on the page, they will be clickable through the transperent page, and trigger the page below. In addition, if `ZoomOnPageClick` feature is enabled (`ZoomOnPageClick="true"`, default false), zooming must be disabled if a button is hovered. To fix these issues, a short script must be added to the page's root to control buttons:

```
function reset(onTop,active) {  
    button1.enabled = onTop;  
    button2.enabled = onTop;  
    //etc.  
}
```

```
//for each button:
//needed only if ZoomOnClick feature is enabled
button1.onRollOver = function () {
pfroot.overbutton = true;
}
button1.onRollOut = button1.onDragOut = function () {
pfroot.overbutton = false;
}
button1.onRelease = function () {
pfroot.overbutton = false;
}
//etc.
```

Please note, that if you have already defined a reset function, just add the script enclosed by the braces to the existing reset function.

Animation/Sound/Video on the page

Animations, Sound and Video playback are also controlled by the reset function. If a page is not visible, the animation or playback on it must be paused, and resumed when it become visible again. To control it, add this script inside the existing reset function:

```
if(onToplactive) {
//play script
} else {
//pause script
}
```

Embedded pages

Page files can be embedded in pageflip_v36 fla. Make your own page MovieClip, then export it using the library's linkage setting (export it to the first frame). The Export Name have to be used to reference this page in the XML file: `PageFile="embedded:yourPageMC"` Use prefix: "embedded:" in any PageFile and LargeFile to tell the script that the content is embedded. (Thumbnails can't be embedded, use LiveThumbnails)

zoom

Each page can have an associated zoom file, that will loaded when the zoom button is pressed. if ZoomOnClick is enabled, it is enough to click on the page. The zoom page scrolling can be done by dragging it (`DragZoom="true"`, default false) or by mouse movement over the page. The size of the area where zoom pages are loaded can be changed also (if PageScale is enablet it will be scaled like the book), that is important if you change the Stage size.

thumbnails

If `LiveThumbnails` option is disabled, each page must have an associated thumbnail file, they will be loaded when the thumbnail button is pressed first. They are fixed sized. If `PageScale` is enabled, the thumbnail won't re-scaled, but they will be rearranged to fit the available space on the stage.

Customising the control bar

There is three variables in pageflip_loader fla root script:

`var ControlBarVisible = true;` if don't need the control bar, set it to false

`var ControlBarPosition = "BOTTOM";` or "TOP": sets the position of the control bar

`var DefaultControlHeight = 60;` defines the control bar height. Needed for properly display PageFlip.

Location of the customisable items: pageflip_loader fla -> library -> Customisable items

Symbols name ending with _btn are the button MovieClips, each with (except autoflip) 3 frames -> normal/hovered/pressed.

ctrl is the layout MovieClip. Make attention of the script in that MovieClip, it controls the functionality of the buttons. If you change the buttons, its MovieClip must have the same instance name as it has originally.

Customising the Stage

If the pages are not correctly fit on the stage, you have to set [BaseStageWidth](#) and [BaseStageHeight](#) attributes in the XML file matching with the size of the pageflip_loader.fla document's Stage. The control-bar and the pages will be placed automatically.

Load PageFlip into your own Movie

Copy and paste the Pageflip Loader MovieClip from the pageflip_loader.fla's library to your own Movie (in this case, page scaling won't work correctly).

Setting up the XML file

The Dynamic PageFlip 3.5 uses an XML configuration file to store preferences and content information. This XML file can be edited with a text editor (not word processor!) like TextEdit/TextWrangler/NotePad or can be generated by a server-side script.

The XML filename can be set in Pageflip Loader MovieClip in the library of pageflip_loader.fla.

Save PageFlip files (XML file included) into the same folder as the pageflip_loader.swf, or in a sub-folder of that folder

Settings

The first part of the XML configuration file, used to configure the book.

PageWidth="330" PageHeight="380" AutoSize="true"	Set the page size in pixel. AutoSize enabled resizes the loaded page which size is different than PageWidth/PageHeight.
StartPage="1"	Sets the starting page number. At the start the book will be opened at this page. This value is overwritten by directlink if it is defined at html's embed level.
AlwaysOpened="false"	If enabled, the book will start opened, and can't closed anymore.
CenterSinglePage="true"	If enabled, the book still centered on the Stage when only one page is visible (first/last pages).
RightToLeft="false"	Display book so its browsing direction is reversed. This option also affects page number display, thumbnail display (only if LiveThumbnails option is disabled), gotoPage function.
VerticalMode="false"	Enable/disable calendar like vertical flipping
OffsetX="0" OffsetY="0"	PageFlip offcenter offset.
PageScale="true" MaxScale="200"	Enable/disable page scaling when the stage is resized. MaxScale can limit upscaling percent of the pages.
BaseStageWidth="720" BaseStageHeight="560"	Sets the default Sage size, on which the PageFlip book is shown at 100% scale. Take effect only if PageScale is enabled.
Hardcover="true" Hardpage="false"	PageFlip can handle two type of pages, "hard" and "soft", If you set Hardcover to "true" the covers will be hard, if "false" they will be soft. Hardpage do the same with the inside pages.
EmbossedPages="true"	Enables or disables the shine/shadow effect at the center of the book.

LargeCover="true" LargeCoverTopOversize="10" LargeCoverSideOversize="10"	With this option set to "true" you can use larger cover pages as the inside pages. Oversize values tells the script how many the cover pages are larger than inside pages.
BackContent="true" BackFile="pages/demo_back.swf" BackActive="true" BackAlwaysVisible="true" BackContentFixed="false"	BackContent option let you load a MovieClip or image file in the background of the PageFlip. This content will be center aligned with the book. With this option set to "true", a file will be loaded from the URL set in BackFile. BackAlwaysVisible have to be set to false, if the BackContent is covered completely with the opened book. Otherwise it must still visible even if the book is opened.
BackgroundColor="0xF0F0F0" buttonColor="0x333333" buttonOverColor="0xF0B400" buttonPressColor="0x333333"	Set up the Background colour, and the colours of the buttons in different states
startAutoFlip="false" AutoFlipDefaultInterval="5" AutoFlipLooping="true"	As the book is loaded, it starts flipping automatically through the whole book. The default values for the interval and looping will be used for autoflipping.
TransparencyEnabled="true" TransparencyActivePage="true"	Enables/disables the 1 level transparency in the book. If you don't use transparent pages, set both value to "false" to use less CPU. Set TransparencyActivePage to "true" if you want active animations on the page covered by the transparent page.
FullScreenEnabled="true"	Enables/disables the FullScreen button on the control bar.
PDFLink="test.pdf"	Links the defined file to the Book button on the control bar.
ButtonHelp="true"	If "true" it shows a little help message when a button is hovered.
ZoomOnClick="true" LightBoxZoom="true" ZoomActivePage="true" DragZoom="false" ZoomToTop="true" ZoomHotspot="true"	ZoomOnClick means that a simple click on the page switch to zoom view. LightBoxZoom set to "true" enables zoom images view with LightBox (no swf support). If ZoomActivePage is "true", animations on the zoomed page will be active. If DragZoom is "true" the zoomed page can moved by dragging it. ZoomToTop show the top of the page when entering zoom view. ZoomHotspot enables HotSpots used on the original page in zoom view.
ThumbnailsEnabled="true" LiveThumbnails="true" ThumbnailScale="true" ThumbnailWidth="104" ThumbnailHeight="120" ThumbnailPadding="20" ThumbnailBorderWidth="0" ThumbnailBorderColor="0x000000" ThumbnailBorderHoverColor="0x000000"	Set up of the Thumbnail view layout.
HcoverAngle="-30" ClickArea="64" AutoFlipArea="56" GotoPageSpeed="2" MousePursuitSpeed="5" OnReleaseSpeed="5" TearDistance="80"	Please modify these variables carefully, they affect page flipping animation control. HcoverAngle set the angle of the opened hard pages. ClickArea is the shape width at the outer side of each page, where you can drag to begin flip that page. AutoFlipArea is the box size at the page corners where you can flip a page by simply clicking on it, and it also used for curl the page corner at mouse-over. Speed values controls animation speed at different events. 1 is the fastest, and as you increment this value the animation is slower. TearDistance defines how far you need to drag a page to tear it off.
FlippingEnabled="true"	Simply enables/disables the ability to flip pages.
PageCache="3"	Defines the number of pages (back and forward) to be pre-loaded from the actual page.
UnloadPages="true"	If enabled, unused pages are unloaded, so they will free up some memory.

<code>PageLoaderBackColor="0x333333"</code> <code>PageLoaderColor="0x666666"</code>	Defines the colours used on the empty page that is only visible before the page file was loaded or in case of that file is not accessible. It uses Hexadecimal RGB values.
<code>MouseControl="true"</code>	Enables/disables mouse control of the book.

Sounds

PageFlip sounds are controlled by the following attributes:

<code>Enabled="true"</code>	Enables/Disables the flipping sounds in PageFlip. It is default false.
<code>EffectVolume="10"</code>	Set the flipping sounds volume from 0 to 100. Default 25.
<code>Start="pf1.mp3"</code> <code>End="pf2.mp3"</code> <code>TearOff="pf3.mp3"</code>	Sound files played at the specified events. Start for start flipping, End for end flipping (when mouse button is released while dragging the page), and TearOff for tear off page sound.

PageOrder

The second part of the XML configuration file. Pages are defined in this section. Each page is defined in a [pagedata](#) tag, and have the following attributes:

<code>PageFile="path/pagefile.jpg"</code>	Defines the path and the file to be loaded for that page. PageFlip supports JPGs, non-animated GIFs, PNGs and SWF file formats.
<code>ZoomFile="path/zoomfile.jpg"</code>	Optional. Defines the zoomed page file, it is loaded when you click on the magnifier button. If you don't define this attribute the zoom buttons will be inactive (except LargeFile is set).
<code>PrintFile="path/printfile.jpg"</code>	Optional. Defines the print page file, it is loaded and printed when you click on the print button. If you don't define this attribute the print buttons will be inactive (except LargeFile is set).
<code>LargeFile="path/largefile.jpg"</code>	Optional. It defines the Zoom and Print files at once.
<code>ThumbnailFile="path/thumbnail.jpg"</code>	If ThumbnailsEnabled is true and LiveThumbnails is disabled, ThumbnailFile must be defined for each page.
<code>Label="pagelabel"</code>	Optional. It is used by gotoPageLabel(); and the Table of Contents. You can refer to this page by its Label instead of the page number (which changes if you add or remove a page before this).
<code>TearOff="true"</code>	Optional. Default it is false. If you set it to true, you will be able to tear out that page.
<code>HardPage="true"</code>	Optional. Default it is false. If true, the page will flip like a hard-cover.
<code>HotspotData="path/hotspot.xml"</code>	Optional. This is a hotspot data file, which contains a hotspot map, with popup messages and links
<code>Caption="Zoom Image Caption"</code>	Optional. Used by LightBox Zoom feature, Caption is the subtitle of the zoom-image in LightBox Viewer.
<code>Data="anything"</code>	Optional. A custom variable to send data to a loaded swf page. This variable can be accessed by the script of any Flash page file. You can load the same swf file for many of the pages, and use Data variable to give them different roles or content.
<code>PageName="Cover"</code>	Optional. Add custom page name, to display it in the control bar. A second usage of this attribute is using "*"5" syntax: it let you start page numbering from this page using the

number after asterisk. For example you can name the first two pages "Cover" and give the name "*"1" to the third page will result a book, which page numbering starts on the first page after the cover.

TableOfContents

This part is optional, it helps you build a table of content, that can be accessed from any page of the book. Each TOC item is defined in a [Link](#) tag, and have the following attributes:

Name="Name of TOC item"	The name of that TOC item, as it will appear on the page.
TargetLabel="pageLabel"	TOC item target page is identified by the label set in pageorder.
TargetPage="1"	TOC item target page is identified by its number. If you set it to 9999, the link will open the thumbnail view.
Skip="true"	Optional. Default is "false". If you set it to true, you will go directly to the target page without flipping through the book.

Language

This part is optional, it can change all text displayed by PageFlip:

SoundLoadingText ZoomLoadingText PrintLoadingText PagerText ZoomPagerText ThumbnailPagerSingle ThumbnailPager ThumbnailPreviousPageButtonHelp ThumbnailNextPageButtonHelp PreviousPageButtonHelp NextPageButtonHelp ZoomLeftButtonHelp ZoomRightButtonHelp PrintLeftButtonHelp PrintRightButtonHelp ThumbnailButtonHelp PDFButtonHelp StartAutoFlipHelp StopAutoFlipHelp FullScreenOnButtonHelp FullScreenOffButtonHelp	You can make language translations by redefining each language attributes. Texts that contains a # sign defines the place of a number to be displayed dynamically. For example at page 23 the Pager is displaying "Page 22-23" with the setting PagerText="Page #" , but if set to "#. oldal" then it will display "22-23. oldal"
---	---

Using Developer features

To get most of the PageFlip's power, you certainly will need access to functions and event handlers. Here is a short list of available functions and their syntax:

PageFlip Core

The following functions can be used to control PageFlip (to access these functions from the page file's script, use pfront global variable as prefix).

<code>pfront.gotoPage(TargetPage, Skip);</code>	TargetPage is the page number of the target page Skip can be <code>true</code> or <code>false</code> (default false). If true, gotoPage goes to the target page with one flip, if false (or not defined) all page will be flipped until reaching the target page.
<code>pfront.gotoPageLabel("Label", Skip);</code>	Label is the page label of the target page set in XML Skip can be <code>true</code> or <code>false</code> (default false).
<code>pfront.gotoPrevPage();</code> <code>pfront.gotoNextPage();</code>	Flip to the previous/next page.
<code>pfront.showZoom(PageNumber);</code> <code>pfront.hideZoom();</code>	Show/Hide Zoom view of the page specified in PageNumber.
<code>pfront.showThumbs();</code>	Toggles Thumbnail view. Thumbnail Mode can be also activated by jumping to page 9999 using the <code>pfront.gotoPage(9999);</code> function call.
<code>pfront.startAutoFlip(sec, loop);</code> <code>pfront.stopAutoFlip();</code> <code>pfront.toggleAutoFlip(sec, loop);</code>	Start, Stop or Toggle automatic flipping. During AutoFlip you can't manually control the flipping. Parameter sec is the intervall between two flippings, and loop can be true or false, to enable/disable looping (after the last page, the book opens at first page again)

Useful Variables

The following functions can be used to control PageFlip (to access these functions from the page file's script, use pfront global variable as prefix).

<code>_global.pfront:MovieClip</code>	Use pfront to access main PageFlip functions and variables from anywhere in the MovieClip hierarchy (ex: <code>currentPage = pfront.page;</code>).
<code>page:Number</code>	Page number of the current page (usually returns the left side page number). Read only variable, do not modify.
<code>zoomMode:Boolean</code>	Check this variable to know if you are in Zoom View Mode or not.
<code>thMode:Boolean</code>	Check this variable to know if you are in Thumbnail View Mode or not.
<code>p:Array</code>	Contains all informations about pages, including the target MovieClip that holds the loaded page content.

Need more information?

If you still need help on PageFlip set up, or you think this page is not informative enough, please contact us.