Flipping Book Flash Component Documentation

Version 1.1

Contents

Description

This component is intended for page flipping and supports flash mx and flash mx 2004 IDEs. You may get the component ready to work in just a few seconds right from the development environment. The component supports pages containing library symbols, external JPEG images and external SWF files. Also, flipping sound is supported (the source sound can be loaded as a library symbol or from an external file). To specify pages simply enter the relative URL of the image or the SWF file, or the linkage ID in the movie library.

The page flipping component provides for software page caching. This allows caching control and provides maximum comfort for the users. No more annoying blinks while another page is being loaded from the browser cache.

Where necessary, you may perform all the component settings from an external XML file that supports all the setting capacities of the development environment, for example, change page order or any other parameter - just edit the XML file.

This is the world's only flash development that provides such a great page flipping animation control opportunity. You may define page flipping and return speed and the speed of the gotoPage function. You may set the auto flip to respond to mouse pointer located near a page edge. Or, you may disable any response to mouse clicks leaving only the auto flip working, and create any user interface on the page. Users will not be distracted from the page at every mouse click anymore.

The page flip is known to have high CPU resources requirements, therefore our component was initially designed to save CPU resources. On average, this component works 30% more efficiently than its known counterparts.

If you want to make a nice looking book with front and rear cover pages, you no longer need to create transparent pages: all you need is to control the Always Opened parameter.

In spite of its size, this component has the most powerful API. You may dynamically create component copies, control them during the work and trace, at the flash player level any component generated events. You may extend the component any way you want - your only limit is now your own fantasy.

The component usability is so designed that even someone not familiar with its properties can make a full capacity movie clip with page flipping in just a few seconds. Believe us: having installed the component and prepared the images or animations for the pages you'll create your first movie with page flipping in less than a minute.

The component is delivered in the form of an MXP file so it will be installed on your computer automatically and become available on the components panel.

Features

All features of the object are listed below.

Basic features

- The simplest page adding procedure ever
- Simultaneous support of pages containing library symbols, external SWF files and JPEG images
- Page Preloader (you may edit or disable it as necessary)
- 2 page caching modes (full preloading and loading on demand)
- Support of Macromedia Flash Player Versions 6 and 7
- Possibility of component setting either from the IDE or with an external XML file
- Most powerful extension tools (component API)
- Optimal use of the CPU resources
- Small target file size (10 Kb)

Animation control

- Animation control (page flipping speed, shadows depth and background color)
- 2 imaging styles (with or without cover page support)
- First book page number control
- Fully controlled automatic page flipping (auto flip)
- Pages can be flipped with or without mouse clicking
- Support of any flipping sounds (through an external mp3 file)
- gotoPage function (any page)
- directGotoPage function

Additional features

- Efficient software caching (no reloading or page blinking when flipping - nobody else can provide anything comparable)
- Detailed documentation
- User friendly Live Preview
- Intelligent usability (creation of a standard catalog takes 20-30 seconds)

Documentation

Purchase and Installation

Once you purchase the component, the link to the download file will be sent to you by e-mail. This will be a zip archive with the component files, the documentation and a set of flip sounds. Download this archive to your computer, unzip it and run the flippingBook.mxp file. This will install the component on your computer. If you are not currently running the FLASH, close and restart it. The next time the environment is started the component will appear in the components panel and be ready to work.

Starting the Work

To start working with the component simply drag its copy onto the Stage and specify page content. For high quality motion this effect usually requires high fps values. In our examples we used fps = 30.

Parameter Description

Book Pages

This parameter is a set of paths to the component page content sources. The order of pages is determined by the order the paths are specified. The paths can be library link IDs and paths to external swf and jpg files. If a path is incorrect, an empty page will be created.
Any book contains an even number of pages, therefore if you specify an odd number of pages, an empty page will be added in the end of the book. **The pages are numbered beginning from 0.**

First Page Number

Number of the page from which the book will be opened upon the start of playing the movie.

Always Opened

This parameter determines the appearance of the book. If its value is True, the book is always opened, if False, you may add the front and rear cover pages and make your book open and close.

Autoflip Area Size

This parameter determines the size of the area near the book edges for automatic flip starting upon bringing the mouse pointer to that area. To disable this effect, set this parameter to 0.

Flip on Click

This parameter determines whether flipping will be started by mouse clicking a page. If set to False, flipping will not start.

Page Moving Speed

This parameter sets the relative page moving speed during flipping. The higher its value the higher the speed and lower the page moving smoothness. The default value is 2.

Page Closing Speed

This parameter sets the relative page moving speed during closing.

GotoPage Speed

This parameter sets the relative page moving speed during the gotoPage(n) operation.

Shadows Depth

This parameter sets the shadows depth in pages. If set to 0, no shadows appear.

Flip Sound

This parameter is the path to the flip sound source file. Its value may be either an external mp3 file path or a movie library link ID.

Page Background Color

This parameter determines the page background color. The color will be displayed when loading pages and used as color for empty pages. The color should be set in RGB in the following format: 0xNNNNNN, where N is a hexadecimal number (0-F).

Load Pages on Demand

This parameter controls page caching. If set to True, a page will be loaded only when the user browses all the previous ones. If set to False, all the pages will be loaded upon the start of the component.

Cache Pages

This parameter determines the page caching necessity. If set to True, all the pages will be saved in the component software cache and not reloaded on next demand. If set to False, pages will be reloaded on every demand. This allows creating books with dynamic pages. The default value is True.

Use Preloader

This parameter determines the necessity of using the preloader during page loading. If set to True, the preloader will be used, otherwise it will not be sued. The preloader is only used for page loading. You may edit the preloader by customizing its appearance. To do this open the FB Components\Premium pageflip component\PagePreloader directory in the movie library, select the preloader symbol and edit it. Ensure that the scaled component has the name bar_mc.

External XML Source

This parameter sets the name of the external XML file with the component settings. If no name is specified, the development environment settings will be used. If the name is incorrect, the component will not work.

XML file format

The component settings XML file is a usual text file of the following format:

```
<FlippingBook>
        <width>200</width>
        <height>100</height>
        <firstPage>5</firstPage>
        <alwaysOpened> true </alwaysOpened>

        <autoFlip> 50 </autoFlip>
        <flipOnClick> true </flipOnClick>
        <shadowsDepth> 2 </shadowsDepth>
        <moveSpeed> 5 </moveSpeed>
        <closeSpeed> 3 </closeSpeed>
        <gotoSpeed> 3 </gotoSpeed>

        <flipSound> mp3/sound1.mp3 </flipSound>
        <pageBack> 0xB4DBF5 </pageBack>
        <loadOnDemand> true </loadOnDemand>
        <cachePages> true </cachePages>
        <usePreloader> true </usePreloader>
        <pages>
               <page> img/01.jpg </page>
               <page> swf/my_movie.swf</page>
               <page> swf/login_form.swf</page>
               <page> Symbol1</page>
        </pages>
</FlippingBook>
```

The embedded XML analyzer of the component is case and odd space insensitive, so be accurate but not afraid of making a mistake. As can be seen, all the file information is contained in the **`<FlippingBook></FlippingBook>`** structure. This is the core description unit - do not modify it. This unit contains component parameter units the names and values of which are the same as those of the component parameters in the development environment.
The Width and Height parameters allow controlling component sizes in the movie. The general parameter value format is as follows:

```
<parameter name>parameter value</parameter name>
```

The only exception is the set of pages by **`the <pages></pages>`** unit. This unit contains the list of pages in the following form: **`<page>page path</page>`**. The order of pages is determined by the order of page source paths. Remember that the source data can be not only external photo files, but also library symbols and external SWF files.

Component API

Component methods

- **setSize(width, height)**
  Sets book size

  example:
  instanceName.setSize (200, 250);

  instanceName is the name of component instance on the stage.

- **gotoPage (n)**
  Goes to page n. You can use it only when book is not in flipping process.

  example:
  instanceName.gotoPage(12);

- **flipForward()**
  Flips one page forward. You can use it only when book is not in flipping process.

example:
instanceName.flipForward();

- **flipBack()**
Flips one page back. You can use it only when book is not in flipping process.

  example:
  instanceName.flipBack();

- **directGotoPage(n)**
Goes to page n directly. Skipping unnecessary pages flipping. You can use it only when book is not in flipping process.

  example:
  instanceName.directGotoPage(12);

Component properties

- **autoFlipProp**
Get/Set Autoflip Area Size property

  example:
  instanceName.autoFlipProp = 50;

  instanceName is the name of component instance on the stage.

- **flipOnClickProp**
Get/Set Flip on Click property

  example:
  instanceName.flipOnClickProp = true;

- **moveSpeedProp**
Get/Set Page Moving Speed property

  example:
  instanceName.moveSpeedProp = 2;

- **closeSpeedProp**
Get/Set Page Closing Speed property

  example:
  instanceName.closeSpeedProp = 3;

- **gotoSpeedProp**

  example:
  instanceName.gotoSpeedProp = 4;

- **alwaysOpenedProp**
Get alwaysOpened property. Read only.

  example:
  var tmp = instanceName.alwaysOpened;

Component events

- **onClick(i)**
  Appears only when the user mouse clicks on page i

  example:
  ```
  instanceName.onClick = function(i)
  {
  	trace("Page clicked: " + i);
  }
  ```

  instanceName is the name of component instance on the stage.

- **onPutPage(i, page)**
  Appears after flipping page i. You can access page movie clip using **page** param.

  example:
  ```
  instanceName.onPutPage = function(i, page)
  {
  	trace("Page number: " + i);
  	page.gotoAndPlay("anyFrame");
  }
  ```

- **onLastPage()**
  Appears when end of book reached

  example:
  ```
  instanceName.onLastPage = function()
  {
  	trace("End");
  }
  ```

- **onFirstPage()**
  Appears when beginning of book reached

  example:
  ```
  instanceName.onFirstPage = function()
  {
  	trace("Start");
  }
  ```

  Note: first page means page #0.

- **onStartFlip(i)**
  Appears when page i flipping starts

  example:
  ```
  instanceName.onStartFlip = function(i)
  {
  	trace("Page #" + i + " is flipping");
  }
  ```

- **onFlipBack(i)**

Appears when page i is returned back following the user releasing the mouse button

example:

```
instanceName.onFlipBack = function(i)
{
        trace(“Page #” + i + “ is returned”);
}
```

- **onEndGoto()**

  Appears when the gotoPage function is complete

  example:

  ```
  instanceName.onEndGoto = function()
  {
          trace(“End of gotoPage procedure”);
  }
  ```

- **onPageLoad(url)**

  Appears when page loading is complete.

  example:

  ```
  instanceName.onPageLoad = function(url)
  {
          trace(“Page: ” + url + “ loaded”);
  }
  ```

Tutorials

Creation of a Simple Photo Catalog

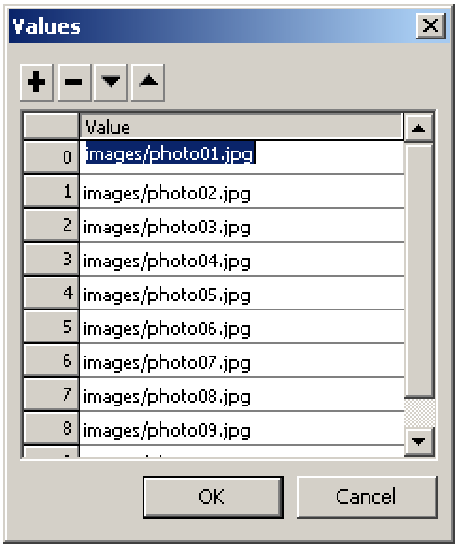
In this Lesson we will consider in detail the creation of a simple photo catalog using the page flip component.

1. First, select the photos you wish to include into your catalog. The component supports JPEG image file format. If your photos are in another format, first export them to JPEG with any suitable graphic editor. To get the best image quality, ensure your photos are of the same size as the component page. This will avoid distortion and save CPU resources. Photo filenames may include any text, but for the sake of definitiveness we will assume that you have 10 photos and named them photo01.jpg, photo02.jpg - photo10.jpg.

2. Create the FBookLesson directory on your computer and then the Images directory therein. Save your photos to the Images directory.

3. Start the flash MX or MX 2004 (or 2004 MX professional) development environment with the page flip component installed. Open the components panel and drag the Premium Pageflip component onto the Stage.

4. Click it to get access to the settings. In the Component Properties panel (usually in the bottom of the Stage), you will see all the component parameters. First there comes the Pages parameter that contains paths to all the media sources for the pages. Mouse click on this parameter and enter the paths to your photos in the dialog opened. In our example these will be images/photo01.jpg, images/photo02.jpg - images/photo10.jpg. Press OK.



	Value
0	images/photo01.jpg
1	images/photo02.jpg
2	images/photo03.jpg
3	images/photo04.jpg
4	images/photo05.jpg
5	images/photo06.jpg
6	images/photo07.jpg
7	images/photo08.jpg
8	images/photo09.jpg

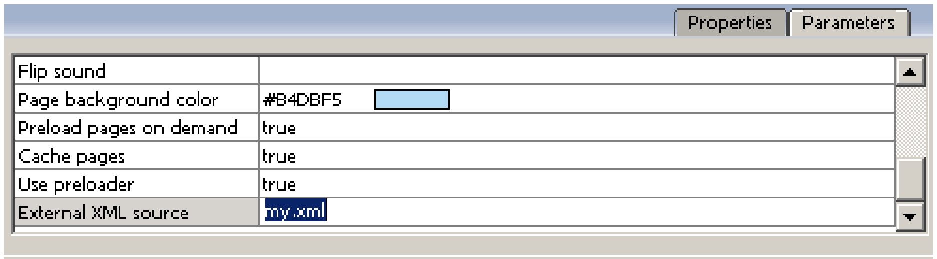
5. Save your fla file to the previously created FBookLesson directory as lesson01.fla and start playing your movie. Congratulations! The job is complete! Page flipping already works excellently. If you wish to additionally customize the component to suit your specific needs, use other component parameters.

Note: media sources can be not only photo file paths, but also SWF file paths and library symbol paths (to use this option specify the symbol link ID as the source).

Creation of a Photo Catalog Using XML

In this Lesson we will create a movie clip using the page flip component identical to the one described in the first Lesson. The only difference will be that the component will receive its setting parameters not from the development environment component parameters but from an external XML file.

1. Open the FLA file created in Lesson No. 1 and save it as lesson02.fla. Select the copy of the component on the Stage to get access to its properties. Browse the parameters list to the bottom and enter my.xml for the External XML parameter. Do not start playing your movie - we haven't yet created the configuration xml file:



Flip sound	
Page background color	#B4DBF5 🟦
Preload pages on demand	true
Cache pages	true
Use preloader	true
External XML source	my.xml

2. To create it, use any text editor like notepad, create a text file with the following content:

```
<FlippingBook>
               <width>200</width>
               <height>100</height>

               <firstPage>5</firstPage>
               <alwaysOpened> true </alwaysOpened>
               <autoFlip> 50 </autoFlip>
               <flipOnClick> true </flipOnClick>
               <shadowsDepth> 2 </shadowsDepth>
               <moveSpeed> 5 </moveSpeed>
               <closeSpeed> 3 </closeSpeed>
               <gotoSpeed> 3 </gotoSpeed>

               <flipSound> mp3/sound1.mp3 </flipSound>
               <pageBack> 0xB4DBF5 </pageBack>

               <loadOnDemand> true </loadOnDemand>
               <cachePages> true </cachePages>
               <usePreloader> true </usePreloader>
               <pages>
                      <page> images/photo01.jpg </page>
                      <page> images/photo02.jpg </page>
                      <page> images/photo03.jpg </page>
                      <page> images/photo04.jpg </page>
                      <page> images/photo05.jpg </page>
                      <page> images/photo06.jpg </page>
                      <page> images/photo07.jpg </page>
                      <page> images/photo08.jpg </page>
                      <page> images/photo09.jpg </page>
                      <page> images/photo10.jpg </page>
               </pages>
</FlippingBook>
```

and then save it to the earlier created FBookLesson directory as **my.xml**. (Be careful: text editors often add the .txt extension to the files they have been used to create. All the XML file parameters correspond to the component parameters. Their description and additional information are available in the Documentation Section. Make sure the target file has exactly the name my.xml. The component is not sensitive to file extension, but the file name should be the same as the value of the External XML Source parameter.

3. Now go back to the development environment and start paying the movie. Done!

Support

We tried to make the component reliable, simple and easily understood. However, there may be an unlimited number of different questions and wishes of users. This section is intended to help our users with their problems.

Contact us: **support@flippingbook.com**