Help file for

ELEGANT PHOTOGRAPHY TEMPLATE

By ICEONFLAMES @ FLASHDEN

Table of Contents

[The package 5](#_Toc233175197)

[Introduction 10](#_Toc233175198)

[Minimum requirements to run the template 10](#_Toc233175199)

[Installation 10](#_Toc233175200)

[Security (very important…please read) 10](#_Toc233175201)

[Organization of the template 11](#_Toc233175202)

[Basic HOW-TO’s 11](#_Toc233175203)

[Change the logo 12](#_Toc233175204)

[Change the music file 12](#_Toc233175205)

[Change the title of the browser window 12](#_Toc233175206)

[Change the “***© Your name”*** in the context menu of the template 12](#_Toc233175207)

[Change the background slideshow images 12](#_Toc233175208)

[Change the content of the ABOUT and NEWS sections 12](#_Toc233175209)

[Add/remove/edit the menu items 13](#_Toc233175210)

[Change images for galleries 13](#_Toc233175211)

[Change images and edit login for login galleries 13](#_Toc233175212)

[Change the quotes that appear on the home page 13](#_Toc233175213)

[Edit the styles for the text loaded in content pages 13](#_Toc233175214)

[The XML files 13](#_Toc233175215)

[siteconfig.xml 13](#_Toc233175216)

[sitemainmenu.xml 14](#_Toc233175217)

[The <item> tag 14](#_Toc233175218)

[The unexplained tags 15](#_Toc233175219)

[home.xml 18](#_Toc233175220)

[homeslideshow.xml 19](#_Toc233175221)

[The <options> tag 19](#_Toc233175222)

[The <images> tag 20](#_Toc233175223)

[The unexplained tags 20](#_Toc233175224)

[XML files for content pages 22](#_Toc233175225)

[XML files for normal galleries 23](#_Toc233175226)

[portraits.xml 23](#_Toc233175227)

[infoheight 23](#_Toc233175228)

[rows 23](#_Toc233175229)

[gap 23](#_Toc233175230)

[scaletostage 23](#_Toc233175231)

[galwidth 23](#_Toc233175232)

[galheight 24](#_Toc233175233)

[homeimg 24](#_Toc233175234)

[scalemode 24](#_Toc233175235)

[zoom 24](#_Toc233175236)

[mousescroll 24](#_Toc233175237)

[gallery1.xmL 24](#_Toc233175238)

[thumbwidth and thumbheight 25](#_Toc233175239)

[Columns 25](#_Toc233175240)

[gap 25](#_Toc233175241)

[XML files for login galleries 26](#_Toc233175242)

[clients.xml 26](#_Toc233175243)

[clientgalleries.xml 26](#_Toc233175244)

[The FLA files 27](#_Toc233175245)

[contact.fla 27](#_Toc233175246)

[encrypt.fla 27](#_Toc233175247)

[gallery.fla 27](#_Toc233175248)

[home.fla 27](#_Toc233175249)

[keyboard.fla 28](#_Toc233175250)

[logingallery.fla 28](#_Toc233175251)

[main.fla 28](#_Toc233175252)

[md5.fla 28](#_Toc233175253)

[preloader.fla 29](#_Toc233175254)

[The HTML site 29](#_Toc233175255)

[Requirements 29](#_Toc233175256)

[The files involved 29](#_Toc233175257)

[How it all works 29](#_Toc233175258)

[The XSL Files 30](#_Toc233175259)

[sitemainmenu.xsl 30](#_Toc233175260)

[text.xsl 30](#_Toc233175261)

[core.xsl 30](#_Toc233175262)

[images.xsl 31](#_Toc233175263)

[htmlpage.php 32](#_Toc233175264)

[I don’t want to generate the HTML page 32](#_Toc233175265)

[Google Analytics 32](#_Toc233175266)

[Other things you may want to know 33](#_Toc233175267)

[New features 33](#_Toc233175268)

[Updates 33](#_Toc233175269)

[More help 33](#_Toc233175270)

[com.maheshj.utils package 33](#_Toc233175271)

[ContentLoader.as 33](#_Toc233175272)

[ImageLoader.as 33](#_Toc233175273)

[IXML.as 33](#_Toc233175274)

[com.maheshj.utils.mp3 package 34](#_Toc233175275)

[MP3PlayBackManager.as 34](#_Toc233175276)

[PlayList.as 34](#_Toc233175277)

[PlayListItem.as 34](#_Toc233175278)

[com.maheshj.UI package 34](#_Toc233175279)

[MouseScrollBox.as 34](#_Toc233175280)

[RMenu.as 34](#_Toc233175281)

[ScrollBar.as 34](#_Toc233175282)

[ScrollBox.as 34](#_Toc233175283)

[VerticalScrollBox.as 35](#_Toc233175284)

[Links and references 35](#_Toc233175285)

[The End 35](#_Toc233175286)

# The package

The downloaded package contains the following files and folders

* *[Folder]*as
  + *[Folder]*bgswitcher
    - BGImage.as
    - BGSwitcher.as
    - BGVideo.as
    - Content.as
    - SwitcherPanel.as
    - TextContentLoader.as
    - Thumb.as
    - ThumbList.as
  + *[Folder]*caurina (Tweener classes)
  + *[Folder]*com
    - *[Folder]*maheshj
      * *[Folder]*UI
        + MouseScrollBox.as
        + RMenu.as
        + ScrollBar.as
        + ScrollBox.as
        + VerticalScrollBox.as
        + *[Folder]*core

Dimensions.as

DisplayObject.as

GenericMenu.as

GenericMenuItem.as

GenericMenuManager.as

* + - * *[Folder]*utils
        + ContentLoader.as
        + ImageLoader.as
        + IXML.as
        + *[Folder]*mp3

Mp3PlayBackManager.as

PlayList.as

PlaylistItem.as

* + - *[Folder]*pixelbreaker (MacMouseWheel classes)
  + *[Folder]*imagegal
    - BgImage.as
    - BottomMenu.as
    - BottomMenuItem.as
    - Content.as
    - MainMenu.as
    - MainMenuItem.as
    - TextContentLoader.as
    - Thumb.as
    - ThumbList.as
    - XMLImageGallery.as
  + *[Folder]*template
    - Contact.as
    - Footer.as
    - Gallery.as
    - GlobalVars.as
    - LoginGallery.as
    - MainMenu.as
    - MainMenuItem.as
    - MainMenuManager.as
    - MenuItemInitData.as
    - MP3Player.as
    - Page.as
    - PhotogQuotes.as
    - PhotographerTemplate.as
    - StageManager.as
    - TextContent.as
    - TextContentLoader.as
  + EncryptDecrypt.as
  + MD5.as
  + MD5Login.as
  + SWFAddress.as
  + SWFAddressEvent.as
* *[Folder]*fla
  + contact .fla
  + custom.fla
  + encrypt.fla
  + gallery.fla
  + home.fla
  + keyboard.fla
  + logingallery.fla
  + main.fla
  + md5.fla
  + preloader.fla
* *[Folder]*help
  + *[Folder]*apidoc
  + Help.pdf
  + Description.pdf
* *[Folder]*images
  + *[Folder]*big
  + *[Folder]*thumbs
* *[Folder]*xml
  + about.xml
  + clientgalleries.xml
  + clientgalleriesunencrypted.xml
  + clients.xml
  + clientsunencrypted.xml
  + contact.xml
  + contentpagetemplate.xml
  + gallery1.xml
  + home.xml
  + homehomeslideshow.xml
  + nature.xml
  + news.xml
  + people.xml
  + portraits.xml
  + siteconfig.xml
  + sitemainmenu.xml
* *[Folder]*js
  + expressinstall.swf
  + swfobject.js
  + swfaddress.js
  + swfmacmousewheel.js
* *[Folder]*music
  + 1.mp3
* *[Folder]*music
  + *[Folder]*big
  + *[Folder]*thumbs
* *[Folder]*fonts
  + pf\_tempesta\_seven.ttf
  + gnuolane free.ttf
  + SCRIPTIN.ttf
* *[Folder]*xsl
  + core.xsl
  + images.xsl
  + sitemainmenu.xsl
  + text.xsl
* clientemail.php
* contact.swf
* custom.swf
* email.php
* encrypt.swf
* gallery.swf
* home.swf
* html.php
* htmlpage.php
* index.php
* keyboard.swf
* logingallery.swf
* main.swf
* md5.swf
* nohtml.html
* Photographer's template.as2proj
* Preloader.swf
* serverinfo.php
* statichtml.html
* styles.css
* textstyle.css

If you find any discrepancies in the files, you can contact FlashDen support (<http://www.flashden.net/support>) or mail me fom my profile page (<http://www.flashden.net/user/iceonflames>).

# Introduction

First of all, thanks for buying the template. I hope it will help as you anticipated in your projects. If you have any queries, or want to contact me for a freelance project or just to say hey...you contact me from my profile page (<http://www.flashden.net/user/iceonflames>).

# Minimum requirements to run the template

If you want to have the auto generated HTML website, then you need to have PHP 5.0 or above installed on your server. It should also have XSL support. If you have to know if your server supports these features, open the ***serverinfo.php*** from your browser. The path may be something like ***http://www.yourwebserver.com/templatepath/serverinfo.php***. Once the page is displayed, you can get all the information about your server. Once done, it is better to delete the file from your server to prevent unauthorized people from opening it.

# Installation

For installation, first your server needs to fulfill the requirements listed above. Then upload the following files and folders to the root directory of your website to install it the way it is when you downloaded it from flashden.

* *[Folder]*images
* *[Folder]*js
* *[Folder]*music
* *[Folder]*videos
* *[Folder]*xml
* *[Folder]*xsl
* All PHP files
* All SWF files except encrypt.swf, md5.swf and keyboard.swf
* styles.css and textstyle.css

# Security (very important…please read)

As you might already know (from the description of the template I gave on Flashden), the ***CLIENT LOGIN*** section uses MD5 encryption and substitution ciphers to protect the data of your clients (names, passwords, email id’s) which is stored in the XML files. Also, the decryption takes place in the swf files. There are commercially available decompilers out there that can decompile the swf files and hence anybody can get the algorithm used to encrypt the XML files (the password is much safer though, as it uses MD5 hashes). So if security of the images you display in this section is a really big issue to you, then you should consider other alternatives like databases (a separate implementation from what I supplied in this template). Another alternative is to use Amayeta SWF encrypt (www.amayeta.com/software/swfencrypt/) to encrypt your swf files to prevent them from being decompiled.

# Organization of the template

As you might already know, the template is fully XML driven with respect to its settings, content, etc. As far as the file structure is concerned, as you might expect once you look into the files, the code (actionscript) files are located in the ***as*** folder. All the flash source files (.fla) are located in the ***fla*** folder, all the images in the ***images*** folder, music files in the ***music*** folder, videos in the ***videos*** folder, XML files in the ***xml*** folder and finally the XSL files in the ***xsl*** folder. The remaining PHP, SWF and HTML files are placed in the root directory.

It will be really easy to set up your project if you have FlashDevelop installed (just in case you want to modify the code). FlashDevelop is a really powerful actionscript editor. Get it at [www.flashdevelop.org](http://www.flashdevelop.org). If you have FlashDevelop installed on your computer, you can open the **Photographer's template.as2proj** to start editing the code. You can use Flash also to edit the actionscript files, but FlashDevelop has many more useful features like code hinting, etc.

If you don’t want to have a HTML site auto generated, you can use the **nohtml.html**. Just rename it to **index.html** and rename **index.php** to something else.

If you make any changes to the website content and want to see how the HTML website looks, you can open the **html.php** from your browser to view the HTML site.

You need not host the **encrypt.swf**, **md5.swf** and **keyboard.swf** files on your server. In fact, I recommend that you don’t host them. I’ll explain the use of these files later in this help file.

**Note:** Before opening the flash files to edit, install the fonts supplied in the **fonts** folder.

# Basic HOW-TO’s

Below I’ve enlisted some basic things that you might want to customize about the template to get you started.

## Change the logo

There are two places where you need to update the logo – ***preloader.fla*** and ***main.fla***. In ***preloader.***fla, from the library (CTRL + L), open ***logo*** movieclip and put in your own logo. Similarly, in ***main.fla***, open the library and edit ***logo*** movieclip. Save and publish both the documents.

## Change the music file

There are two ways of doing this.

1. Open ***siteconfig.xml*** and change the ***<track>*** tag’s value to your own mp3 file’s path.
2. Replace ***1.mp3*** in the ***music*** folder with your own music file and rename it as ***1.mp3***.

To know more about how to use ***siteconfig.xml***, refer [here](#_siteconfig.xml).

## Change the title of the browser window

First of all to change the title, you need to set the title in ***<head>*** tag in ***index.php***.

Then open ***xml->siteconfig.xml*** and edit the two tags for variables ***sitetitle1*** and ***sitetitle2***. If you take a closer look at how the template works, when you click on one of the menu items (say ABOUT), the the browser title changes to “: : : PHOTOGRAPHER TEMPLATE – ABOUT : : :”. So basically the browser title is made up of ***sitetitle1*** + ***selected menu item title*** + ***sitetitle2***. So select your title accordingly.

## Change the “***© Your name”*** in the context menu of the template

Open ***xml->siteconfig.xml*** and edit the ***<item>*** tags of variables ***siteowner*** and ***ownerlink***.

## Change the background slideshow images

You need to edit the ***homeslideshow.xml*** file to add/remove/change images and other settings of the slideshow. To know more about the XML file, refer [here](#_homeslideshow.xml).

## Change the content of the ABOUT and NEWS sections

You have to edit the ***about.xml*** and ***news.xml***. To know more about it, refer [here](#_XML_files_for).

## Add/remove/edit the menu items

Edit ***sitemainmenu.xml***. You can know more about it [here](#_sitemainmenu.xml).

## Change images for galleries

See [here](#_XML_files_for_1).

## Change images and edit login for login galleries

See [here](#_XML_files_for_2).

## Change the quotes that appear on the home page

If you don’t find it really useful to display the quotes on photography, you can always put in other info in the slideshow like recent news, links to latest projects and galleries (the template has deeplinking…remember!), etc.

## Edit the styles for the text loaded in content pages

You’ll have to either edit the ***textstyle.css*** or apply supported HTML styles in the XML file itself. So basically you need to have an understanding of supported CSS and HTML tags in Flash text.

# The XML files

You’ll find all the XML files in the ***xml*** directory. You can see a lot of them in the folder. Below I explain the use of each of them in the template

## siteconfig.xml

Use this file to set up the initial settings for the template. These are the settings required to load all the components of the template that require external files and settings correctly.

Each ***<item>*** tag assigns properties to either a variable defined in ***PhotographerTemplate.as*** or a movieclip in the ***main.fla***. Below give is its structure.

<item mc=”my\_mc”>

<setting1>setting1\_value</setting1>

<setting2>setting2\_value</setting2>

</item>

<item variable=”vatiablename”>

<value>variable\_value</value >

</item>

Currently in the siteconfig.xml, you’ll find settings for three movieclips namely, ***bg\_mc***, ***menu\_mc*** and ***music\_mc***. ***bg\_mc*** and ***menu\_mc*** actually have an actionscript property called ***xmlFile*** which is required for the BG Switcher and site menu to work. Similarly, ***music\_mc*** (the MP3 player) has a property called ***track*** which is the path to the MP3 file that is to be played.

So as you might understand, if you need to assign any property value to any movieclip inside the root timeline of the site, you can do so in the ***siteconfig.xml*** file.

## sitemainmenu.xml

This file holds the settings for the menu of the template. The ***gap*** attribute in the ***<mainmenu>*** tag is the gap between adjacent menu items.

### The <item> tag

The following table describes all the tags in the *options* tag. Tags descriptions marked with (\*) will be explained later in this document.

|  |  |
| --- | --- |
| **Tag** | **Explanation** |
| <title>HOME</title> | The title of the menu item. This is what is displayed for the item in the template |
| <slideshow>true</slideshow> | \*Whether the BG slideshow should play when this item is loaded or not |
| <ID>home</ID> | A unique ID for the menu item. This is used by SWFAddress to change the URL in the browser. This is also used in setting up the anchor(<a>) tags for each section in the HTML site. |
| <type>file</type> | \*The type of file that is being loaded. Valid values are ***file***, ***contentfile*** and ***text*** |
| <file>home.swf</file> | The file that is to be loaded when the menu item is clicked. This tag is used only when the ***<type>*** is set to either ***<file>*** or ***<contentfile>*** |
| <xmlfile>xml/home.xml</xmlfile> | Once the file is loaded, if it needs to load an XML file, you can use this tag to give it the XML file’s path. Of course you can load the XML file directly in the actionscript, but if you implement your files according to this method, you will have the flexibility to change the XML files’ path on the fly without opening the FLA. |
| <isdefault>true</isdefault> | This is used to set a default item for the menu. This menu item will be directly loaded without clicking it when the template loads. This is optional though. So if you don’t want to load any menu item when the template loads, just don’t use this tag for any menu item |
| <inhtml>false</inhtml> | \*Indicates whether this menu item must be included in the HTML site that is auto generated |
| <inflash>false</inflash> | \*Indicates whether this menu item must be included in the flash template |
| <iscustom>true<iscustom> | \*Indicates whether this menu item must be treated as a custom swf file or not. |

### The unexplained tags

The tags with the (\*) in the above table are explained below.

#### The <slideshow> tag

As mentioned in the table above, this tag is used to denote whether the BG slideshow should play when this item is loaded or not.

To explain further on how this tag works, if the BG slideshow is already playing and this tag is set to ***false***, then the slideshow is paused temporarily and stays that way until another menu item which has this tag set to ***true*** is selected. But if the BG slideshow is already switched off, this setting does not have any effect on it.

The reason why I provided this setting is to save you a bit of bandwidth and load the content faster. For content like galleries, which actially take up the fullscreen when loaded, do not require the slideshow to be running as it cannot be seen. So I put this tag’s value to ***false*** for all the galleries. Similarly you can use this setting to decide if you need the slideshow or not when you load your own custom swf files.

#### The <type> tag

The type of file that is being loaded. Valid values are ***file***, ***contentfile*** and ***text.***

If you want to load an external swf file, set the tag value to ***file***.

If you want to load an external swf file for which you want to display the back BG (click on the ***CONTACT ME*** menu item for example), set the tag value to ***contentfile***.

If you want to load text, set the tag value to ***text***.

Any other values that you want to use will not work.

#### The <inflash> and <inhtml> tags

Use these tags to determine if the menu item will be displayed in the flash and HTML templates or not.

Consider this example tag below:

<item>

<title>GALLERY</title>

<ID>gallery</ID>

<type>file</type>

<xmlfile>xml/nature.xml</xmlfile>

<inhtml>true</inhtml>

<inflash>false</inflash>

</item>

The above menu item will be displayed in the HTML site but not in the flash site.

#### The <iscustom> tag

To understand this tag, you first need to understand how the template handles page loads. If you take a closer look at the main.fla (which is the core of the template), double-click on the ***site\_mc*** movieclip and you’ll see a layer named ***page*** in it, on which there is a movieclip named ***page\_mc***. The code for this movieclip is located in ***Page.as*** file.

It loads the external swf when ***loadFile*** method is called into the ***holder\_mc*** movieclip which actually is of type ***DisplayObject*** (***com.maheshj.UI.core.DisplayObject***). Once the content is loaded, ***page\_mc*** registers itself as a listener to this loaded movieclip for the events ***DisplayObjectInit,*** ***DisplayObjectEnter*** and ***DisplayObjectExit***. The loaded swf must have three methods ***init***, ***enter*** and ***exit***. The init method should dispatch a ***DisplayObjectInit*** event, the enter method should dispatch a ***DisplayObjectEnter*** event and the exit method should dispatch a ***DisplayObjectExit*** event.

When a menu item is clicked, the template first calls the ***exit*** method of ***page\_mc*** so that it will execute the ***exit*** method of the loaded swf file. This is required if you want your swf file to implement a custom ‘exit’ transition (basically you code your exit transition in the ‘exit’ method). Once the exit transition is done, the externally loaded swf dispatches the ***DisplayObjectExit*** event due to which ***page\_mc*** does the cleanup (unloads the movieclip) and dispatches a ***DisplayObjectExit*** event to which the main template responds by executing its own listener (the ***DisplayObjectExit*** method) and loads the selected menu item. Once the item is loaded and page\_mc registers all the listeners, it calls the init method of the loaded swf, in which you can implement all the initialization of the file and dispatch the ***DisplayObjectInit*** event. Once this event is dispatched, page\_mc calls the enter method of the loaded swf in which you can write the code for the “in” transition of the swf and dispatch ***DisplayObjectEnter*** event once the transition is done. This is when ***page\_mc*** assigns the xml file that is to be loaded to the swf by calling the ***xmlFile*** property of the swf and then calling its ***loadXML*** method. For a proper understanding of how this all works, you can refer to ***Gallery.as*** which handles all the functionality of the galleries.

So basically if you want your external swf’s to work with xml files loaded from the menu items xml and have custom “in” and “out” transitions, then in the root timeline of your swf, write the following code at the top on first frame

import mx.events.EventDispatcher;

EventDispatcher.initialize(this);

function init(){

//do all your initialization

//dispatch init event

dispatchEvent({type:”DisplayObjectInit”});

}

function enter(){

//do all your enter transitions

//dispatch enter event

dispatchEvent({type:”DisplayObjectEnter”});

}

function exit(){

//do all your exit transitions

//dispatch exit event

dispatchEvent({type:”DisplayObjectExit”});

}

var xmlFile:String;

function loadXML(){

//execute your XML loading and processing routines here

}

But if you are not comfortable with all these procedures or you are not knowledgeable in event handling procedures of Flash, that is where the ***<iscustom>*** tag comes in. All you have to do is set it to true for that particular menu item in the XML file and the template will do a simple fade in/fade our transition (each lasting for 0.35 seconds) for the external swf you are loading for that menu item. That is what I have done for the ***custom.swf*** file. If you take a look at the code inside ***custom.fla***, you’ll see that I haven’t included any of the code above in it and hence the template does a simple fade in/fade out transition to load and unload it.

## home.xml

Use this file to edit the contents of the “text slideshow” displayed in the ***HOME*** section of the template.

The ***<home>*** tag has an attribute called ***interval*** which is actually the time between each quote in milliseconds.

The ***<quote>*** tag has two tags

* ***author*** – The author of the quote. Appears at the bottom left
* ***text*** – The actual quote

## homeslideshow.xml

This file is fed as the content for the background switcher.

The xml file has two main tags: *options* and *images*.

### The <options> tag

The following table describes all the tags in the *options* tag. Tags descriptions marked with (\*) will be explained later in this document.

|  |  |
| --- | --- |
| **Tag** | **Explanation** |
| <slideshowon>true</slideshowon> | Whether slideshow is on at the beginning of the application – valid values are *true*  and *false* |
| <slideshowint>5</slideshowint> | The time interval for the slideshow in seconds |
| <swalign>right</swalign> | Alignment of the thumbnail list (switcher panel). Valid values – *right*, *left*, *top* and *bottom* |
| <swalignmargin>125</swalignmargin> | \*The align margin |
| <swscroll>vertical</swscroll> | The scroll direction. Valid values - *horizontal*, *vertical* |
| <swdim>500</swdim> | \* The dimension of the switcher panel |
| <swtnum>2</swtnum> | \* Thumbnail number factor |
| <swthumbgap>4</swthumbgap> | The gap between thumbnails in pixels |
| <swthumbwidth>200</swthumbwidth> | \*The width of the thumbnail |
| <swthumbheight>125</swthumbheight> | \*The height of the thumbnail |
| <fadeeasetype>easeInQuart</fadeeasetype> | \*The fade in/out ease type of the bg images |
| <fadetime>0.6</fadetime> | The fade in/out time of the bg images |
| <prevnextdistance>300</prevnextdistance> | Distance of the prev/next buttons from the preloader |
| <infobtndistance>600</infobtndistance> | Distance of the ‘info’ button from the preloader |
| <infodialogwidth>500</infodialogwidth> | Width of the info dialog box |
| <infodialogheight>300</infodialogheight> | Height of the info dialog box |

### The <images> tag

This tag contains a number of *image* tags. Each *image* tag has the following tags:

|  |  |
| --- | --- |
| **Tag** | **Explanation** |
| <title><![CDATA[Title 1]]></title> | The title of the image. If you don’t want to set the title for a particular image, then just don’t write this tag. In that case, the info button becomes invisible |
| <desc><![CDATA[This is description 1.</desc> | The description for the image. If you don’t want to set the description for a particular image, just don’t write this tag. |
| <type>image</type> | Can take values ‘image’ or ‘video’ based on what you want to load |
| <thumb>images/thumbs/1.jpg</thumb> | the path of the thumbnail |
| <img>images/big/1.jpg</img> | the path of the image |

### The unexplained tags

Some tags that were unexplained in the above table are explained here.

#### swalignmargin

The switcher panel (the thumbnails) is located at a distance from the top. This value can be set in XML with this tag. The switcher panel can be located at four different locations on the screen – top, bottom, left and right. For each of these settings this value becomes:

For top, bottom – distance from the left side of the stage (browser or screen)

For right, left – distance from the top of the stage

#### swdim

This value is the width of the switcher panel if the scroll direction is set to *horizontal* and height of the panel if the scroll direction is set to *vertical*.

#### swtnum

This value is the number of rows of thumbnails in the switcher panel if the scroll direction is set to *horizontal* and number of columns of thumbnails in the panel if the scroll direction is set to *vertical*.

#### swthumbwidth and swthumbheight

These values form the width and height of the thumbnails. You must however note that these values are not used to resize the thumbnail images. They are used for the purpose of aligning the thumbnails properly. All you have to do is specify the dimensions of the thumbnails you are using here. If you give higher values, the thumbnail images will centre themselves in the given dimensions.

#### fadeeasetype

This can be any easing type supported by Tweener animation library. The supported ease types are listed below.

* linear
* easeInSine
* easeOutSine
* easeInOutSine
* easeOutInSine
* easeInCubic
* easeOutCubic
* easeInOutCubic
* easeOutInCubic
* easeInQuint
* easeOutQuint
* easeInOutQuint
* easeOutInQuint
* easeInCirc
* easeOutCirc
* easeInOutCirc
* easeOutInCirc
* easeInBack
* easeOutBack
* easeInOutBack
* easeOutInBack
* easeInQuad
* easeOutQuad
* easeInOutQuad
* easeOutInQuad
* easeInQuart
* easeOutQuart
* easeInOutQuart
* easeOutInQuart
* easeInExpo
* easeOutExpo
* easeInOutExpo
* easeOutInExpo
* easeInElastic
* easeOutElastic
* easeInOutElastic
* easeOutInElastic
* easeInBounce
* easeOutBounce
* easeInOutBounce
* easeOutInBounce

For more information refer to <http://code.google.com/p/tweener>

## XML files for content pages

If you have to load a content page (***ABOUT***, ***NEWS*** for example), you have to use ***contentpagetamplate.xml***. Just make a copy of it, give it a name you want and replace the “Your content goes here” text in the XML file with your own. Then create a menu item in the ***sitemainmenu.xml***. Set its ***<type>*** tag to ***text*** and ***<xmlfile>*** tag to the path of this XML file.

See the tag for ***ABOUT*** menu item for a working example.

## XML files for normal galleries

Each gallery ***(gallery.swf)*** needs two types of XML files to work. One is the file containing all the categories and the other XML files are the ones that are loaded when a category is selected.

For example, consider ***portraits.xml*** and ***gallery1.xml***. Both of them are explained below.

## portraits.xml

First, let us have a look at the attributes listed in the *mainmenu* tag. They are

* infoheight
* rows
* gap
* scaletostage
* galwidth
* galheight
* homeimg
* scalemode
* zoom
* mousescroll

### infoheight

The height of the info dialog that appears when the ***info*** button is clicked

### rows

The number of rows the menu should appear in

### gap

The horizontal and vertical gaps between the menu items

### scaletostage

Determines if the gallery is to be scaled according to the dimensions of the stage or not. Valid values are true or false.

### galwidth

Width which the gallery should occupy

### galheight

Height which the gallery should occupy

### homeimg

The image that will be loaded in the background when the gallery loads

The attributes ***scaletostage***, ***galwidth*** and ***galheight*** work together. If ***scaletostage*** is true, then ***galwidth*** and ***galheight*** act as margins for the scaling of the gallery according to the stage dimensions. Suppose ***galwidth***=*100* and ***galheight***=*150*, then during runtime, the width of the gallery will be *stagewidth*-*100* and height will be *stagewidth*-*150*.

On the other hand, if ***scaletostage*** is *false*, then the gallery is just scaled to a width equal to ***galwidth*** and height equal to ***galheight***.

Then comes the ***<item>*** tag. It has three nodes – ***ID, title*** and ***xmlfile***. *ID* is a unique identifier for that particular category. It is used to set the URL of the browser for deeplinking (SWFAddress). titleis the label that appears on the menu item when the gallery loads and ***xmlfile*** is the path of the gallery xml file that is to be loaded when a menu item is selected.

### scalemode

The scale type of the loaded images. Valid values are ***fill***, ***fit*** or ***none***.

### zoom

Option to set if zooming the image with mouse wheel is enabled or not. Set it to ***true*** if you want the zoom to be enabled and vice versa.

### mousescroll

If the loaded image is bigger than the screen dimensions, this option lets the user to scroll the loaded image based on the position on the mouse. Use ***zoom*** and this property to give the visitor a truly interactive experience.

## gallery1.xmL

You can see the following attributes in the file:

* thumbwidth
* thumbheight
* columns
* gap

### thumbwidth and thumbheight

The width and height of the thumb. The above parameters are not used to resize the thumbnails to this size, but to help the gallery align the thumbnails properly. So all you have to do is enter the dimensions of the thumbnails you are using.

### Columns

The number of columns that the thumbnail list should contain

### gap

The gap that must be maintained between the thumbnails

Coming to the *item* tag, it is structured as follows:

<item>

<ID>unique id</ID>

<title>Title text</title>

<desc>Description text</desc>

<thumb>thumbnail path</thumb>

<image>image path</image>

<type>content type</type>

</item>

* ID – the unique identifier for the item. Used to set the browser URL in deeplinking
* title – the title of the image
* desc – the description of the image
* thumb – path to the thumbnail
* image – path to the image
* type – the type of the content to be loaded – valid values are *image* or *swf*. The default type is set to *image*. So if you have to load an image, you don’t have to write this tag. You write it only when you need to load an swf file.

I hope that clears up all the XML part that you need to know. Coming to the flash part, all you have to do is just dive into the gallery.fla and see how everything is organized.

## XML files for login galleries

Here I’m referring to the ***CLIENT LOGIN (logingallery.swf)*** section of the template. You can have as many login galleries as you want.

Coming to the XML files used by this gallery are clients.xml, clientgalleries.xml, and the standard XML files used by gallery.swf (they are described above).

## clients.xml

When you open this file, all you see is random text enclosed in ***<encryptedtext>*** tag. So basically, this is encrypted text.

To view the unencrypted XML data for clients, you can open ***clientsunencrypted.xml***. It is explained below.

Each ***<user>*** tag has the following attributes

* username – the login ID of the client
* password – the password of the client converted to an MD5 hash (I’ll explain this later)
* details – The information in this attribute is used to populate the ***SUBMIT REVIEW*** contact form in the login gallery. Basically it is of the format ***name:(name of the person)/email:(email ID of the person)***.

## clientgalleries.xml

This XML file is also encrypted. So open ***clientgalleriesunencrypted.xml*** file. It is explained below.

Each ***<user>*** tag has the following attributes

* username – the user name of the person
* xmlfile – the xml file that is to be loaded for that particular person. This is a gallery categories xml file.

# The FLA files

These are the heart and soul of the template. All the code for these files are there in the ***template*** folder in ***as*** directory.

## contact.fla

Functionality: The contact form displayed in the ***CONTACT ME*** section

Find code in: ***Contact.as***

Output swf file: ***contact.swf***

## encrypt.fla

Functionality: Use it to encrypt and decrypt the required XML files

Find code in: ***Frame 1*** of ***Actions*** layer.

Output swf file: ***encrypt.swf***

Explanation: You can use the encrypt.swf to encrypt the required XML files. Currently the supported encrypted XML files by the template are ***clients.xml*** and ***clientgalleries.xml***. Open the swf file, give the path of the XML file in the ***XML PATH INPUT*** field (for example xml/clientsunencrypted.xml) and hit the ***ENTER*** on your keyboard. The file is encrypted and displayed in the output below. Copy that to an xml file (in this case clients.xml).

## gallery.fla

Functionality: The galleries displayed in the template

Find code in: ***Gallery.as***

Output swf file: ***gallery.swf***

## home.fla

Functionality: The quotes displayed on the ***HOME*** section

Find code in: ***PhotogQuotes.as***

Output swf file: ***home.swf***

## keyboard.fla

Functionality: Get the code and ASCII value of a particular character on the keyboard

Find code in: ***Frame 1*** in the ***Actions*** layer

Output swf file: ***keyboard.swf***

Explanation: As you might know, the galleries support keyboard shortcuts. If you want to change these shortcuts, you can use this file to get the key code required in the code.

## logingallery.fla

Functionality: The ***CLIENT LOGIN*** section of the template

Find code in: ***LoginGallery.as***

Output swf file: ***logingallery.swf***

## main.fla

Functionality: The main template file

Find code in: ***PhotographerTemplate.as***

Output swf file: ***main.swf***

## md5.fla

Functionality: Use this to create MD5 hashes for you client passwords

Find code in: ***Frame 1*** in ***Actions*** layer in ***md5form*** movieclip

Output swf file: ***md5.swf***

Explanation: If you open the ***clientsunencrypted.xml***, you can see that the passwords are stored as MD5 hashes of the actual password. So you need to use this file to create the MD5 hashes. Also, you should not forget the password or else, there is no way of retrieving it.

## preloader.fla

Functionality: The main preloader of the template

Find code in: Inside the FLA. You’ll have to explore all the movieclips for this one

Output swf file: ***preloader.swf***

# The HTML site

As you already know, this templates creates a hidden HTML page on the fly from the same XML files used to generate the content for Flash site. This HTML page can be used to help search engines index the content of your website.

## Requirements

Your server needs to have PHP 5.0+ and XSLT support. I’ve explained how to check for these in the [beginning](#_Minimum_requirements_to).

## The files involved

The following files are involved in the generation of the HTML page.

* index.php – the main file that is opened when someone visits your website
* htmlpage.php – this file handles the core of generating the HTML page
* xsl files in ***xsl*** folder
* xml files in ***xml*** folder
* Other included files like images, etc.

## How it all works

Basically, I used SWFObject library to embed the swf file into the index.php page. If you open the index.php in a text editor, you can see a ***<div>*** tag with id ***flashcontent***. When SWFObject detects flash player on the visitor’s browser, it replaced the content inside this div tag with the flash file. So when flash player is not present, the content in this tag is displayed. This is where I used the PHP ***require\_once*** function to include ***htmlpage.php*** file.

The ***htmlpage.php*** uses the ***sitemainmenu.xml***, xsl files, styles.css and other XML files mentioned in the ***sitemainmenu.xml*** to generate and format the HTML page. If you want to see the generated HTML page for the default setup you get when you purchased the template, open ***statichtml.html*** in your browser. If you want to see the HTML page generated from the XML files and images you setup for testing purposes, open ***html.php*** in your browser after hosting it on your server (<http://www.yoursitename.com/html.php>).

## The XSL Files

The first question that might pop up in your mind is, “why XSL files?” Well, actually, I can generate the HTML page without using XSL files just by processing the XML files directly in PHP. But using XSL reduces the code a lot as the ***XSLTProcessor*** takes care of the XML processing.

Below is the description of the output from each XSL file present in the ***xsl*** folder.

### sitemainmenu.xsl

* Generates a text output taking content from ... <title>,<id>,{<type>--> 'f':file,'t':text,'l':link},{<xmlfile> for 'f'and 't',<file> for 'l'},\*\*{this serves as seperator}
* Used to generate complete page i.e. links and content
* Used to generate contents of <div id="header"> </div> tag

### text.xsl

* returns complete "text" {CDATA} inside the xml file ... More processing done in php for news.xml
* Used to transform <type>text</type> or <type>contentFile</type> documents.
* Works with about.xml,news.xml,contact.xml

### core.xsl

* returns <title>{CATEGORY #},<xmlfile>{xmlfile for that gallery}
* Works with nature.xml, people.xml, portraits.xml

### images.xsl

* returns one gallery at a time
* depending upon the xml file for gallery passed ...will return the complete gallery like this...
* works with gallery1.xml
* Output resembles the following

{<a href="images/big/1.jpg" target="\_blank"><img src="images/thumbs/1.jpg"/></a><a href="images/big/2.jpg" target="\_blank"><img src="images/thumbs/2.jpg"/></a><a href="images/big/3.jpg" target="\_blank"><img src="images/thumbs/3.jpg"/></a><a href="images/big/4.jpg" target="\_blank"><img src="images/thumbs/4.jpg"/></a><a href="images/big/5.jpg" target="\_blank"><img src="images/thumbs/5.jpg"/></a><a href="images/big/6.jpg" target="\_blank"><img src="images/thumbs/6.jpg"/></a><a href="images/big/1.jpg" target="\_blank"><img src="images/thumbs/1.jpg"/></a><a href="images/big/2.jpg" target="\_blank"><img src="images/thumbs/2.jpg"/></a><a href="images/big/3.jpg" target="\_blank"><img src="images/thumbs/3.jpg"/></a><a href="images/big/4.jpg" target="\_blank"><img src="images/thumbs/4.jpg"/></a><a href="images/big/5.jpg" target="\_blank"><img src="images/thumbs/5.jpg"/></a><a href="images/big/6.jpg" target="\_blank"><img src="images/thumbs/6.jpg"/></a><a href="images/big/1.jpg" target="\_blank"><img src="images/thumbs/1.jpg"/></a><a href="images/big/2.jpg" target="\_blank"><img src="images/thumbs/2.jpg"/></a><a href="images/big/3.jpg" target="\_blank"><img src="images/thumbs/3.jpg"/></a><a href="images/big/4.jpg" target="\_blank"><img src="images/thumbs/4.jpg"/></a><a href="images/big/5.jpg" target="\_blank"><img src="images/thumbs/5.jpg"/></a><a href="images/big/6.jpg" target="\_blank"><img src="images/thumbs/6.jpg"/></a><a href="images/big/1.jpg" target="\_blank"><img src="images/thumbs/1.jpg"/></a><a href="images/big/2.jpg" target="\_blank"><img src="images/thumbs/2.jpg"/></a><a href="images/big/3.jpg" target="\_blank"><img src="images/thumbs/3.jpg"/></a><a href="images/big/4.jpg" target="\_blank"><img src="images/thumbs/4.jpg"/></a><a href="images/big/5.jpg" target="\_blank"><img src="images/thumbs/5.jpg"/></a><a href="images/big/6.jpg" target="\_blank"><img src="images/thumbs/6.jpg"/></a><a href="images/big/1.jpg" target="\_blank"><img src="images/thumbs/1.jpg"/></a><a href="images/big/2.jpg" target="\_blank"><img src="images/thumbs/2.jpg"/></a><a href="images/big/3.jpg" target="\_blank"><img src="images/thumbs/3.jpg"/></a><a href="images/big/4.jpg" target="\_blank"><img src="images/thumbs/4.jpg"/></a><a href="images/big/5.jpg" target="\_blank"><img src="images/thumbs/5.jpg"/></a><a href="images/big/6.jpg" target="\_blank"><img src="images/thumbs/6.jpg"/></a><a href="images/big/1.jpg" target="\_blank"><img src="images/thumbs/1.jpg"/></a><a href="images/big/2.jpg" target="\_blank"><img src="images/thumbs/2.jpg"/></a><a href="images/big/3.jpg" target="\_blank"><img src="images/thumbs/3.jpg"/></a><a href="images/big/4.jpg" target="\_blank"><img src="images/thumbs/4.jpg"/></a><a href="images/big/5.jpg" target="\_blank"><img src="images/thumbs/5.jpg"/></a><a href="images/big/6.jpg" target="\_blank"><img src="images/thumbs/6.jpg"/></a><a href="images/big/1.jpg" target="\_blank"><img src="images/thumbs/1.jpg"/></a><a href="images/big/2.jpg" target="\_blank"><img src="images/thumbs/2.jpg"/></a><a href="images/big/3.jpg" target="\_blank"><img src="images/thumbs/3.jpg"/></a><a href="images/big/4.jpg" target="\_blank"><img src="images/thumbs/4.jpg"/></a><a href="images/big/5.jpg" target="\_blank"><img src="images/thumbs/5.jpg"/></a><a href="images/big/6.jpg" target="\_blank"><img src="images/thumbs/6.jpg"/></a><a href="images/big/1.jpg" target="\_blank"><img src="images/thumbs/1.jpg"/></a><a href="images/big/2.jpg" target="\_blank"><img src="images/thumbs/2.jpg"/></a><a href="images/big/3.jpg" target="\_blank"><img src="images/thumbs/3.jpg"/></a><a href="images/big/4.jpg" target="\_blank"><img src="images/thumbs/4.jpg"/></a><a href="images/big/5.jpg" target="\_blank"><img src="images/thumbs/5.jpg"/></a><a href="images/big/6.jpg" target="\_blank"><img src="images/thumbs/6.jpg"/></a>}

## htmlpage.php

This page combines all the xsl files’ output and generates the HTML page.

## I don’t want to generate the HTML page

Then you can use the ***nohtml.html*** as the main page of your site. Just take the ***index.php*** off the server and rename ***nohtml.html*** as ***index.html***.

# Google Analytics

If you want to plug the template into Google analytics, get the ***ga.js*** code from your analytics account and put it before the ***</body>*** at the bottom of the ***index.php*** page. In the ***<head>*** tag, replace:

<script type="text/javascript" src="js/swfaddress.js"></script>

with

<script type="text/javascript" src="js/swfaddress.js?tracker=pageTracker.\_trackPageview"></script>

# Other things you may want to know

## New features

If you have any new ideas that can become very nice additions to the template, do share them with me so that I can implement them into it.

## Updates

In case I make any updates to the file, like add new features, clear out bugs (if any), there is no way for me to let you know as currently FlashDen doesn’t provide any means to pass on the information to you. So do check back regularly.

# More help

If you need to know more about the available functions, events and properties, please take a look into the *apidoc* folder. Open the *index.html* file in the folder. You might also find the other classes handy in your projects. Some of the useful classes are listed below.

## com.maheshj.utils package

### ContentLoader.as

Use this class to load any content easily without the hassle of writing load event handlers for the MovieClipLoader class.

### ImageLoader.as

Use this class to load dynamically bitmap smoothed images into your projects.

### IXML.as

This is a small utility class to load XML files.

## com.maheshj.utils.mp3 package

Use the classes in this package to code MP3 players that have the design properly separated from the functional logic.

### MP3PlayBackManager.as

Use this class to manage the playback of your mp3 player.

### PlayList.as

Use this class to hold your playlists

### PlayListItem.as

Use this class to hold the data of each track. Your playlist will be made up of instances of this class.

## com.maheshj.UI package

### MouseScrollBox.as

Use this class to implement scrolling content with respect to mouse position. This class is used in my Mouse Content Scroller (<http://www.flashden.net/item/mouse-content-scroller/22078>) file.

### RMenu.as

This class was used in the template to easily manage the right-click menu of the template. You can do the same with this in your projects.

### ScrollBar.as

This class can be used to implement a generic scrollbar that might act as a volume slider or any other slider. It was used in my Easy Content Scroller (<http://www.flashden.net/item/easy-content-scroller-as3/33681>) file.

### ScrollBox.as

This class can be used to implement a content scroller with both side scrolling. It was used in my Easy Content Scroller (<http://www.flashden.net/item/easy-content-scroller-as3/33681>) file.

### VerticalScrollBox.as

While ScrollBox.as can be used to implement just vertical scrolling, I made this class with vertical scrolling functionality to reduce the code and hence the output size.

If you need more information on implementing these classes into your projects, refer to ***index.html*** in ***apidoc*** folder.

# Links and references

SWFObject – <http://blog.deconcept.com/swfobject/>

SWFAddress – <http://www.asual.com/swfaddress/>

Tweener – <http://code.google.com/p/tweener/>

SWFMacMouseWheel – <http://blog.pixelbreaker.com/flash/swfmacmousewheel/>

MD5 algorithm for AS2 – <http://www.sephiroth.it/file_detail.php?id=69>

Public domain images – <http://www.photos8.com>

Event handling in AS2 - <http://www.kirupa.com/developer/actionscript/eventdispatcher.htm>

OOP in AS2 - <http://www.kirupa.com/developer/oop2/AS2OOPindex.htm>

XSLT - <http://www.w3schools.com/xsl/>

CSS - <http://www.w3schools.com/css/>

Free fonts – <http://www.dafont.com>

My Flashden profile – <http://www.flashden.net/user/iceonflames>

My website – <http://www.maheshj.com>

# The End

I’ve hopefully explained all the features of the template and given directions on modifying them. If you have ideas on how this documentation can be improved, please let me know and I’ll do my best.

Rest of the uses is pretty much outlined in the code. If you face any problems implementing it in your projects, do ask... ☺